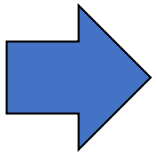
Introduction to Bayesian Statistics - 7

Edoardo Milotti

Università di Trieste and INFN-Sezione di Trieste

Our next important topic: Bayesian estimates often require complex numerical integrals. How do we confront this problem?



enter the Monte Carlo methods!

1. acceptance-rejection sampling
2. importance sampling
3. statistical bootstrap
4. Bayesian methods in a sampling-resampling perspective
5. Introduction to Markov chains and to Random Walks (RW)
6. Simulated annealing
7. The Metropolis algorithm
8. Markov Chain Monte Carlo (MCMC)
9. The Gibbs sampler
10. The efficiency of MCMC algorithms
11. Affine-invariant MCMC algorithms (EMCEE)

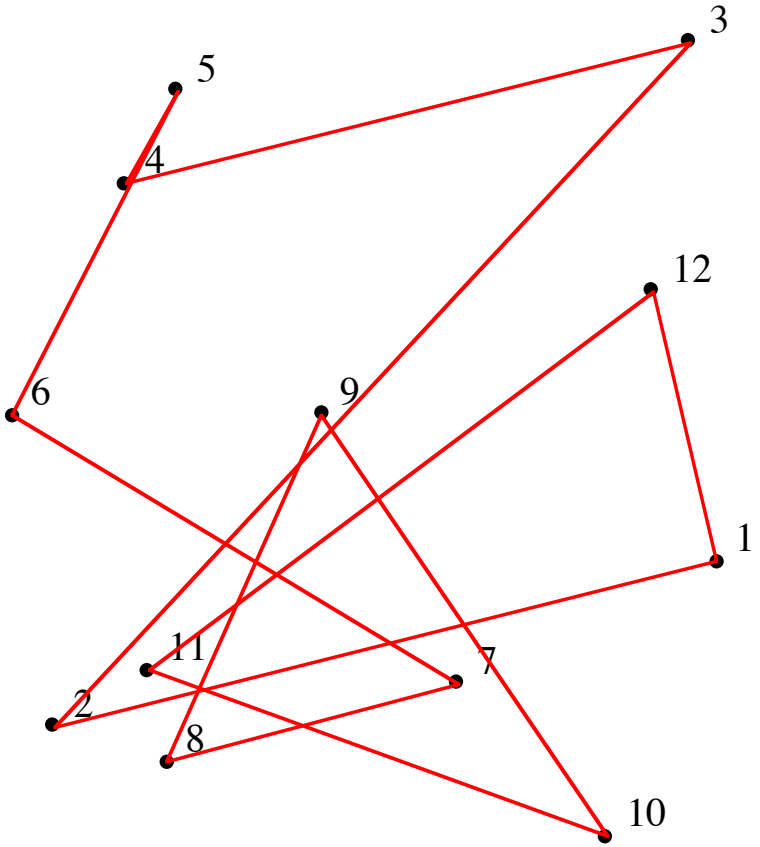
6. The Traveling Salesman Problem and Simulated Annealing

To introduce the method, we consider the *Traveling Salesman Problem* (TSP), where we want to find the shortest closed path that connects N cities.

The problem was first stated by the Viennese mathematician Karl Menger in 1930 and is one of the most studied problems in combinatorial optimization.

For many up-to-date links, see
http://www.math.uwaterloo.ca/tsp/index.html

See also the history page
http://www.math.uwaterloo.ca/tsp/history/index.html



12 “cities” randomly distributed in the (0,1) square: the path corresponds to a random permutation of the sequence of cities.

(path length L=1.93834)

Paths are enumerated by permutations of “city names”, e.g., {9, 2, 7, 8, 1, 12, 4, 5, 3, 10, 11, 6} (start at 9, step to 2, and so on until you reach 6 and then return to 9).

The total number of configurations (undirected paths) is

$$\frac{1}{2}(n-1)!$$

The problem belongs to the class of NP-complete problems (Non-Polynomial complexity, a class of particularly hard problems)

In such cases there is only one known exact solution: the full enumeration of all paths.

13 May 1983, Volume 220, Number 4598 SCIENCE

Optimization by Simulated Annealing

S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi

Summary. There is a deep and useful connection between statistical mechanics (the behavior of systems with many degrees of freedom in thermal equilibrium at a finite temperature) and multivariate or combinatorial optimization (finding the minimum of a given function depending on many parameters). A detailed analogy with annealing in solids provides a framework for optimization of the properties of very large and complex systems. This connection to statistical mechanics exposes new information and provides an unfamiliar perspective on traditional optimization problems and methods.

Approximate solution of the TSP with the Simulated Annealing algorithm

path length → **energy of the system**

exploration of the configuration space with the *Metropolis algorithm*
(Metropolis, Rosenbluth Rosenbluth ,Teller and Teller, 1953)

THE JOURNAL OF CHEMICAL PHYSICS VOLUME 21, NUMBER 6 JUNE, 1953

Equation of State Calculations by Fast Computing Machines

NICHOLAS METROPOLIS, ARIANNA W. ROSENBLUTH, MARSHALL N. ROSENBLUTH, AND AUGUSTA H. TELLER,
Los Alamos Scientific Laboratory, Los Alamos, New Mexico

AND

EDWARD TELLER,* *Department of Physics, University of Chicago, Chicago, Illinois*

(Received March 6, 1953)

A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional rigid-sphere system have been obtained on the Los Alamos MANIAC and are presented here. These results are compared to the free volume equation of state and to a four-term virial coefficient expansion.

7. The Metropolis algorithm and its application to the TSP

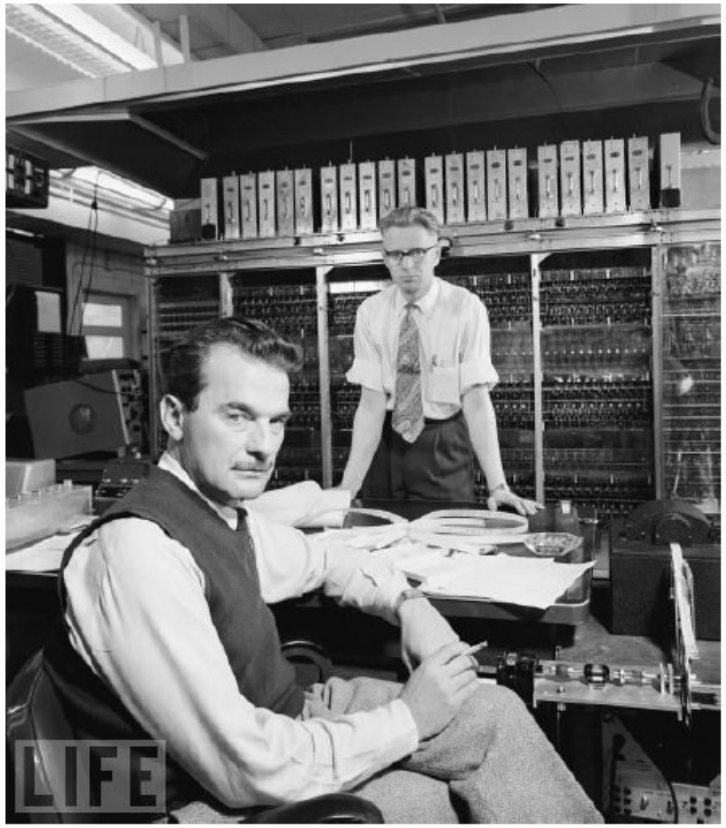


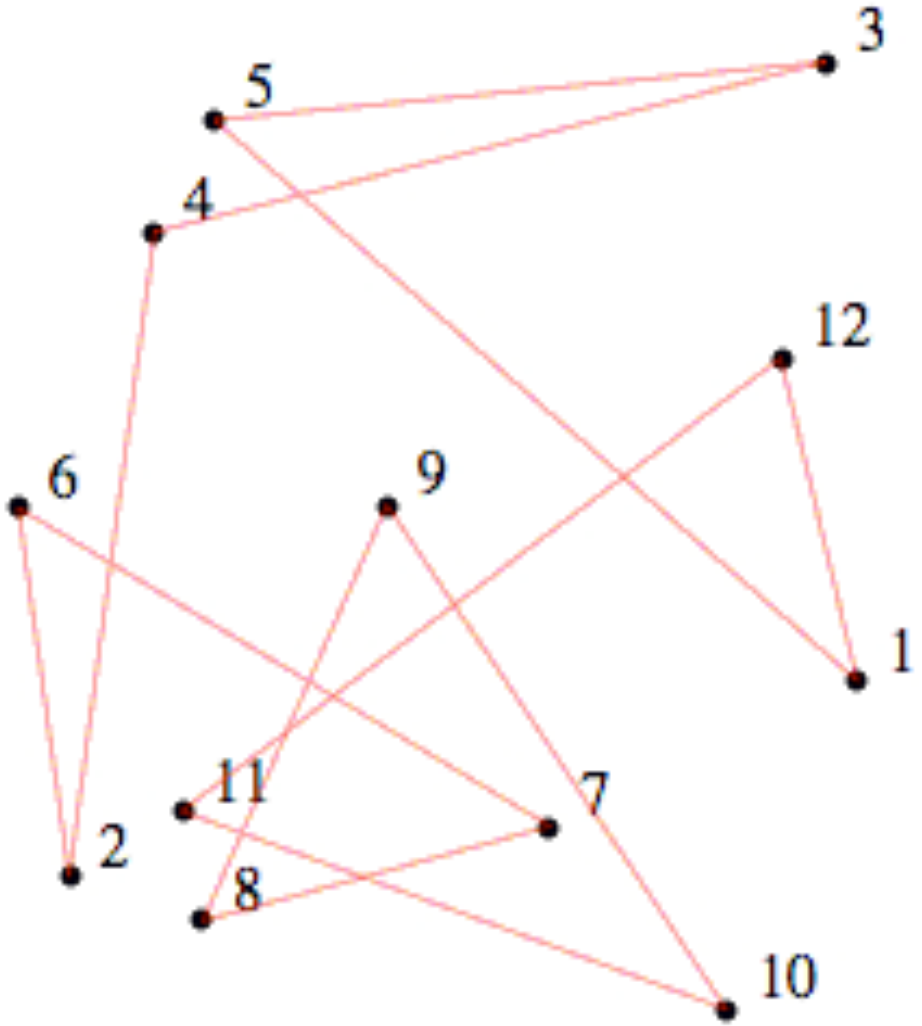
Figure 8.14: Portrait of American computer scientists Nicholas Metropolis (1915 - 1999) (seated) and James Henry Richardson (1918 - 1996) at Los Alamos National Laboratory, Los Alamos, New Mexico, November 1953 (from `http://www.life.com`).

1. We generate a new configuration C′ from the present configuration C
2. We compute the energy of the new configuration, E'
3. We compute the energy difference $\Delta E = E' - E$
4. The new configuration is accepted with probability p

$$\begin{cases} p = 1 & \Delta E < 0 \\ p = \exp\left(-\dfrac{\Delta E}{kT}\right) & \Delta E \geq 0 \end{cases}$$

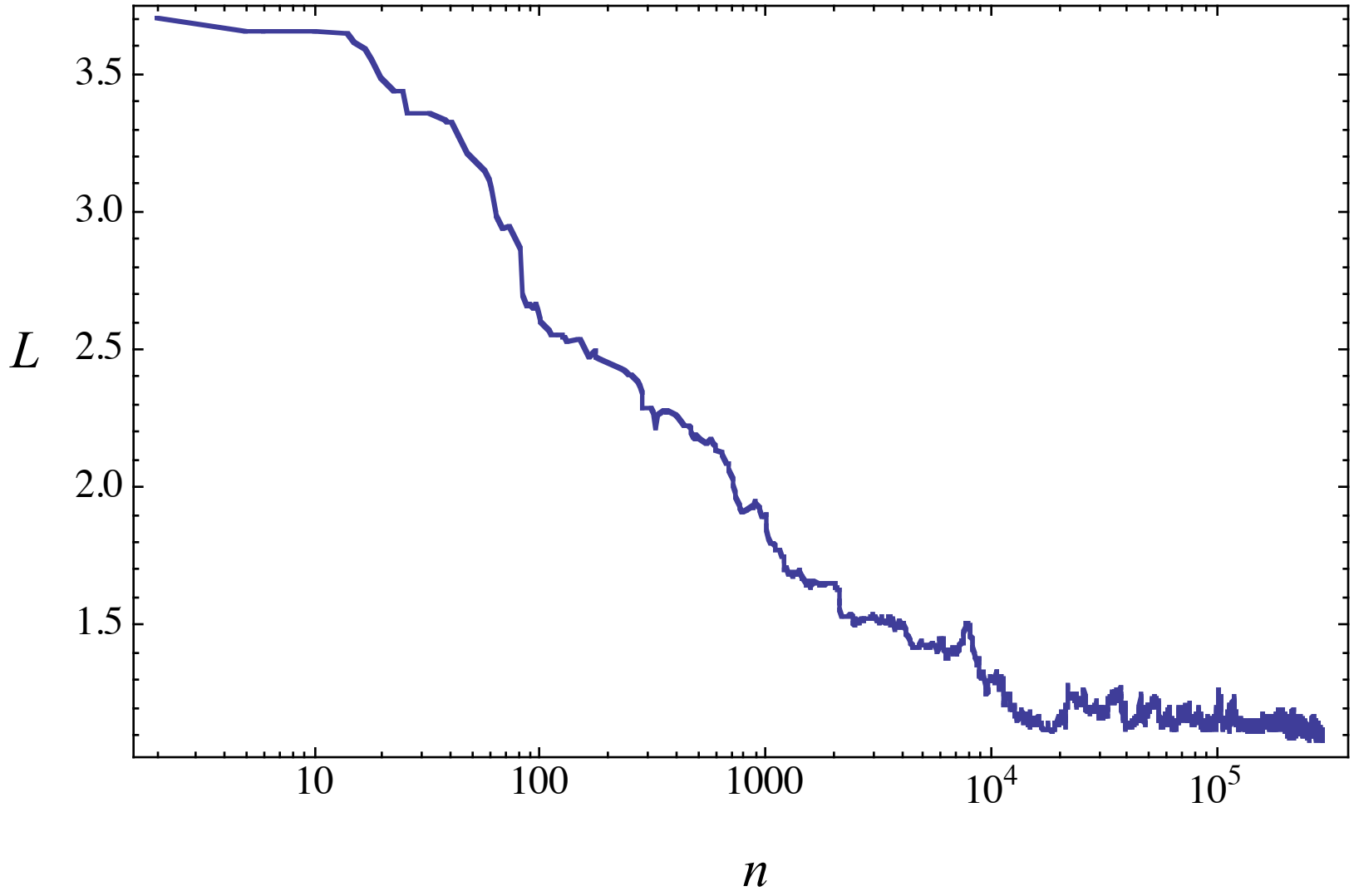
Additional details

- the algorithm needs a slow cooling (it is common to choose an exponential cooling schedule)
- if cooling is not gradual, the system can get stuck into a local minimum
- simple exchanges of pairs of cities are the individual moves in the SA solution of the TSP
- the individual steps from one configuration to the next can be described by a Markov chain



k = 1
T = 0.05
L = 1.84655

Decrease of total path length in a realization of the SA solution of a 50-cities problem



Here we note that the transition probability can be written as follows

$$T\left(C \to C'\right) = \min\left[1, \exp\left(-\frac{\left(E' - E\right)}{kT}\right)\right]$$

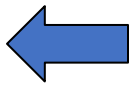
Moreover, it is easy to show that the algorithm preserves detailed balance

$$P(C)T\left(C \to C'\right) = P(C')T\left(C' \to C\right)$$

where P(C) is the stationary probability of configuration C. Indeed, at equilibrium we find that, if E’ > E,

$$P(C)\exp\left(-\frac{\left(E' - E\right)}{kT}\right) = P(C')$$

$$\frac{P(C')}{P(C)} = \exp\left(-\frac{\left(E' - E\right)}{kT}\right)$$



Boltzmann’s distribution

Finally, we can write:

$$T\left(C \to C'\right) = \min\left[1, \frac{P\left(C'\right)}{P\left(C\right)}\right]$$

This definition of the transition probability is the starting point for an important further step, the Metropolis-Hastings algorithm.

8. MCMC – definition of the Metropolis-Hastings (M-H) algorithm (1970)

- we define the transition probability

$$P(\mathbf{x} \to \mathbf{y}) = q(\mathbf{x}, \mathbf{y})\alpha(\mathbf{x}, \mathbf{y})$$

and the target density $\pi(\mathbf{x})$

- we take state $\mathbf{x} = \mathbf{x}_n$

- we choose randomly another state $\mathbf{y}$ and we accept it $\left(\mathbf{y} \to \mathbf{x}_{n+1}\right)$ with probability

$$\alpha(\mathbf{x},\mathbf{y}) = \min\left\{1, \frac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})}\right\}$$

Note that if the proposal function q is symmetrical, then the acceptance probability takes on the simpler form

$$\alpha(\mathbf{x},\mathbf{y}) = \min\left\{1, \frac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})}\right\} \rightarrow \min\left\{1, \frac{\pi(\mathbf{y})}{\pi(\mathbf{x})}\right\}$$

and it depends on the target density only.

The M-H algorithm defines a Markov chain, and it is easy to show that detailed balance holds. The transition probability is

$$P(\mathbf{x} \to \mathbf{y}) = q(\mathbf{x},\mathbf{y})\alpha(\mathbf{x},\mathbf{y}) = q(\mathbf{x},\mathbf{y})\min\left\{1, \frac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})}\right\}$$

- case $\dfrac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})} \geq 1$

$$\Rightarrow \quad \alpha(\mathbf{x},\mathbf{y}) = 1; \quad \alpha(\mathbf{y},\mathbf{x}) = \frac{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})}{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})} \quad \Rightarrow \quad \begin{aligned} P(\mathbf{x} \to \mathbf{y}) &= q(\mathbf{x},\mathbf{y}) \\ P(\mathbf{y} \to \mathbf{x}) &= q(\mathbf{y},\mathbf{x})\frac{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})}{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})} \end{aligned}$$

$$\Rightarrow \quad \begin{aligned} \pi(\mathbf{x})P(\mathbf{x} \to \mathbf{y}) &= \pi(\mathbf{x})q(\mathbf{x},\mathbf{y}) \\ \pi(\mathbf{y})P(\mathbf{y} \to \mathbf{x}) &= \pi(\mathbf{y})q(\mathbf{y},\mathbf{x})\frac{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})}{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})} = \pi(\mathbf{x})q(\mathbf{x},\mathbf{y}) \end{aligned}$$

- case $\dfrac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})} < 1$

$$\Rightarrow \alpha(\mathbf{x},\mathbf{y}) = \frac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})}; \quad \alpha(\mathbf{y},\mathbf{x}) = 1 \Rightarrow \begin{aligned} P(\mathbf{x}\to\mathbf{y}) &= q(\mathbf{x},\mathbf{y})\frac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})} \\ P(\mathbf{y}\to\mathbf{x}) &= q(\mathbf{y},\mathbf{x}) \end{aligned}$$

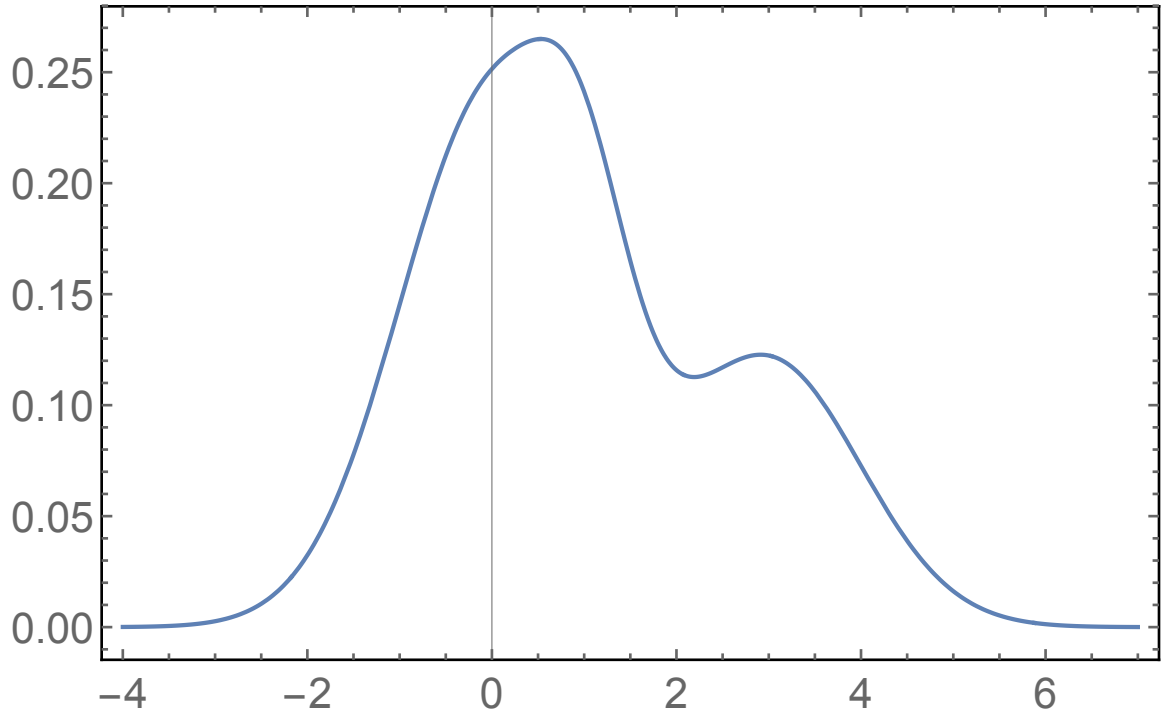
$$\Rightarrow \begin{aligned} \pi(\mathbf{x})P(\mathbf{x}\to\mathbf{y}) &= \pi(\mathbf{x})q(\mathbf{x},\mathbf{y})\frac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})} = \pi(\mathbf{y})q(\mathbf{y},\mathbf{x}) \\ \pi(\mathbf{y})P(\mathbf{y}\to\mathbf{x}) &= \pi(\mathbf{y})q(\mathbf{y},\mathbf{x}) \end{aligned}$$

Detailed balance holds in both cases and therefore $\pi(\mathbf{x})$ is stationary

The following figure shows a simulation with the MCMC algorithm and the distribution

$$p(x) = \frac{0.6}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) + \frac{0.3}{\sqrt{2\pi}} \exp\left(-\frac{(x-3)^2}{2}\right) + \frac{0.1}{\sqrt{0.5\pi}} \exp\left(-\frac{(x-1)^2}{0.5}\right)$$

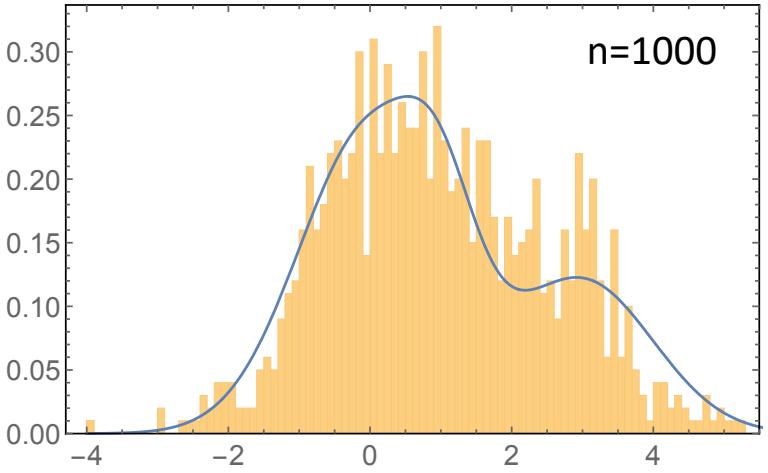
(a three-component mixture model)

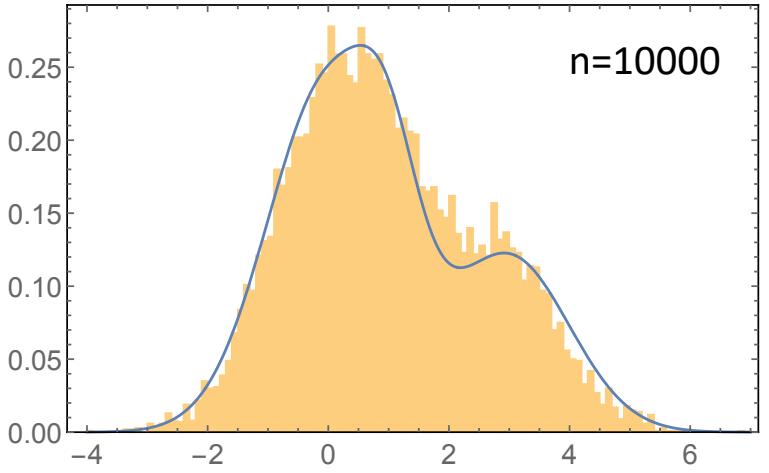


```
nrmax = 40 000;

xr = Table[0, {nrmax}];
xr[[1]] = -4;

nr = 1;
While[nr < nrmax,
 xtry = xr[[nr]] + RandomReal[NormalDistribution[0, 1]];
 If[ pdf[xtry] / pdf[xr[[nr]]] > RandomReal[] , nr++; xr[[nr]] = xtry];
]
```

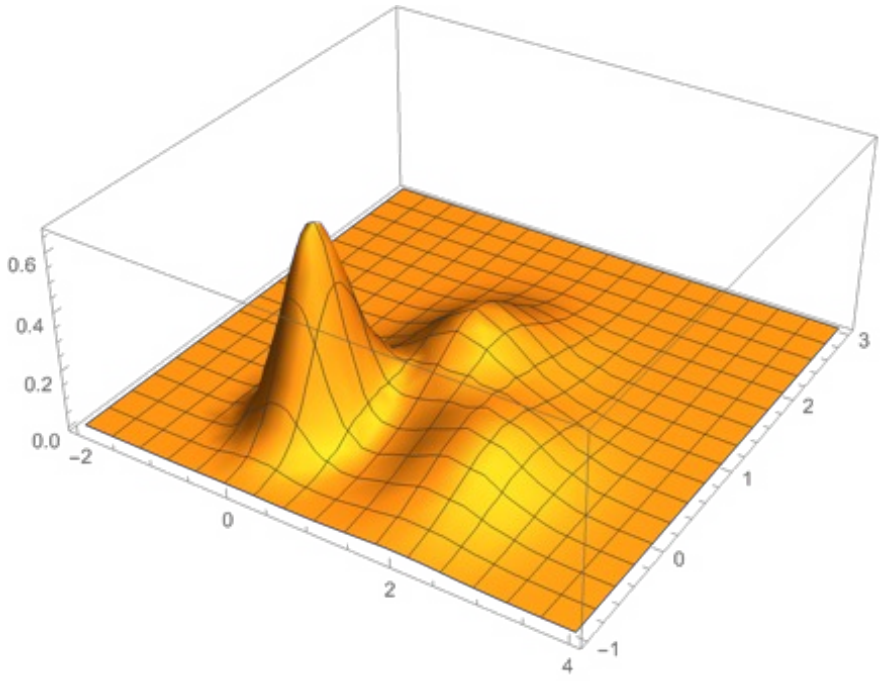


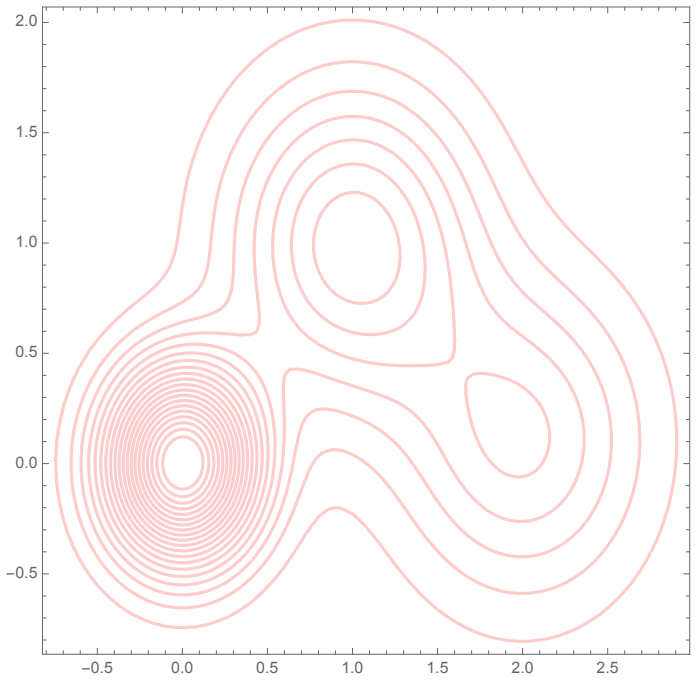


MCMC simulation of a 2D three-component mixture model

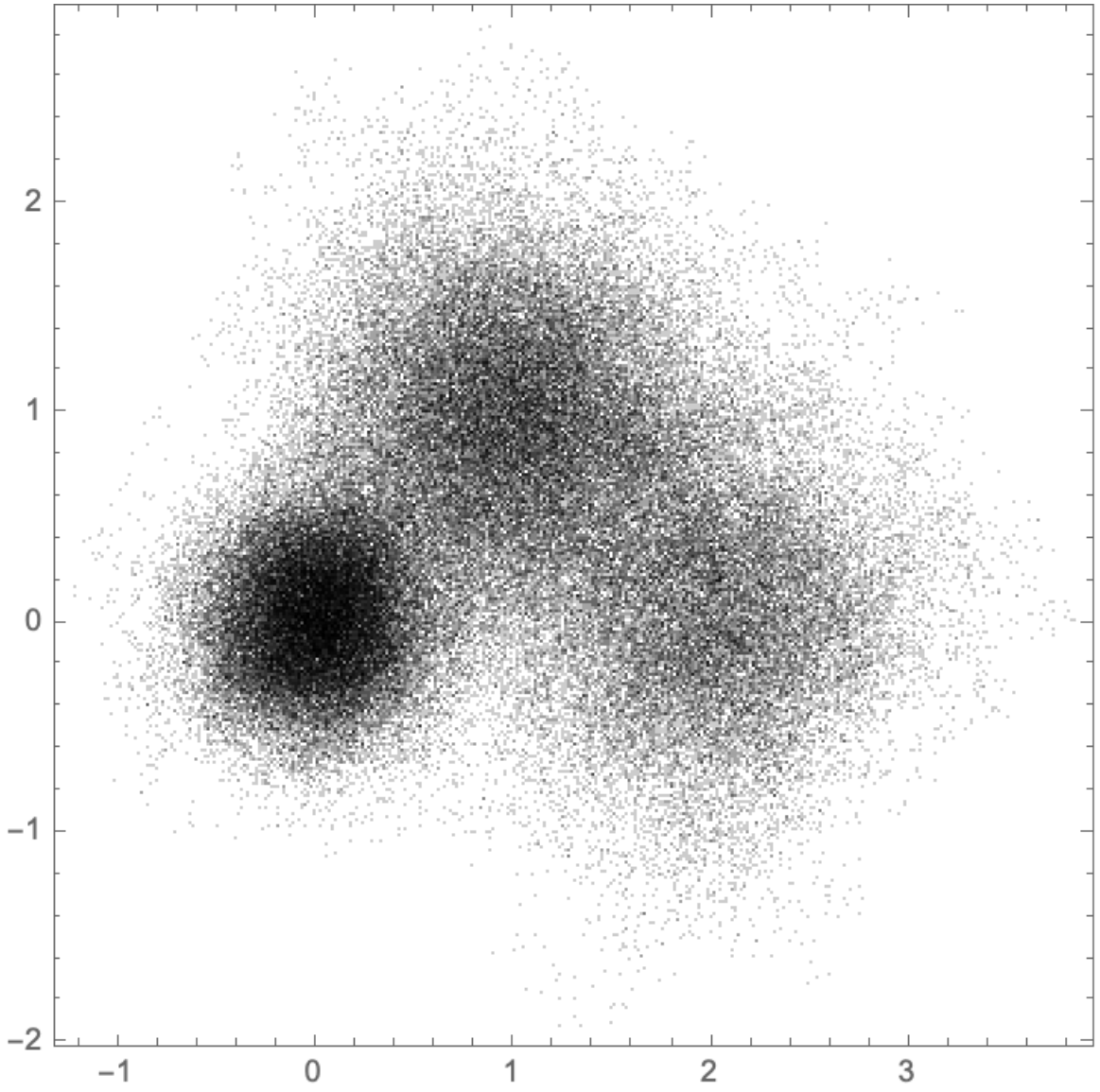
$$p(x,y) = \sum_{i=1}^{3} \frac{\alpha_i}{\sqrt{2\pi\sigma_i^2}} \exp\left[-\frac{(x-\mu_{x,i})^2 + (y-\mu_{y,i})^2}{2\sigma_i^2}\right]$$

$$\begin{array}{llll} \alpha_1 = 0.5; & \mu_{x,1} = 0; & \mu_{y,1} = 0; & \sigma_1 = 0.3; \\ \alpha_2 = 0.3; & \mu_{x,2} = 1; & \mu_{y,2} = 1.; & \sigma_2 = 0.5; \\ \alpha_3 = 0.2; & \mu_{x,3} = 2; & \mu_{y,3} = 0.1; & \sigma_3 = 0.5; \end{array}$$

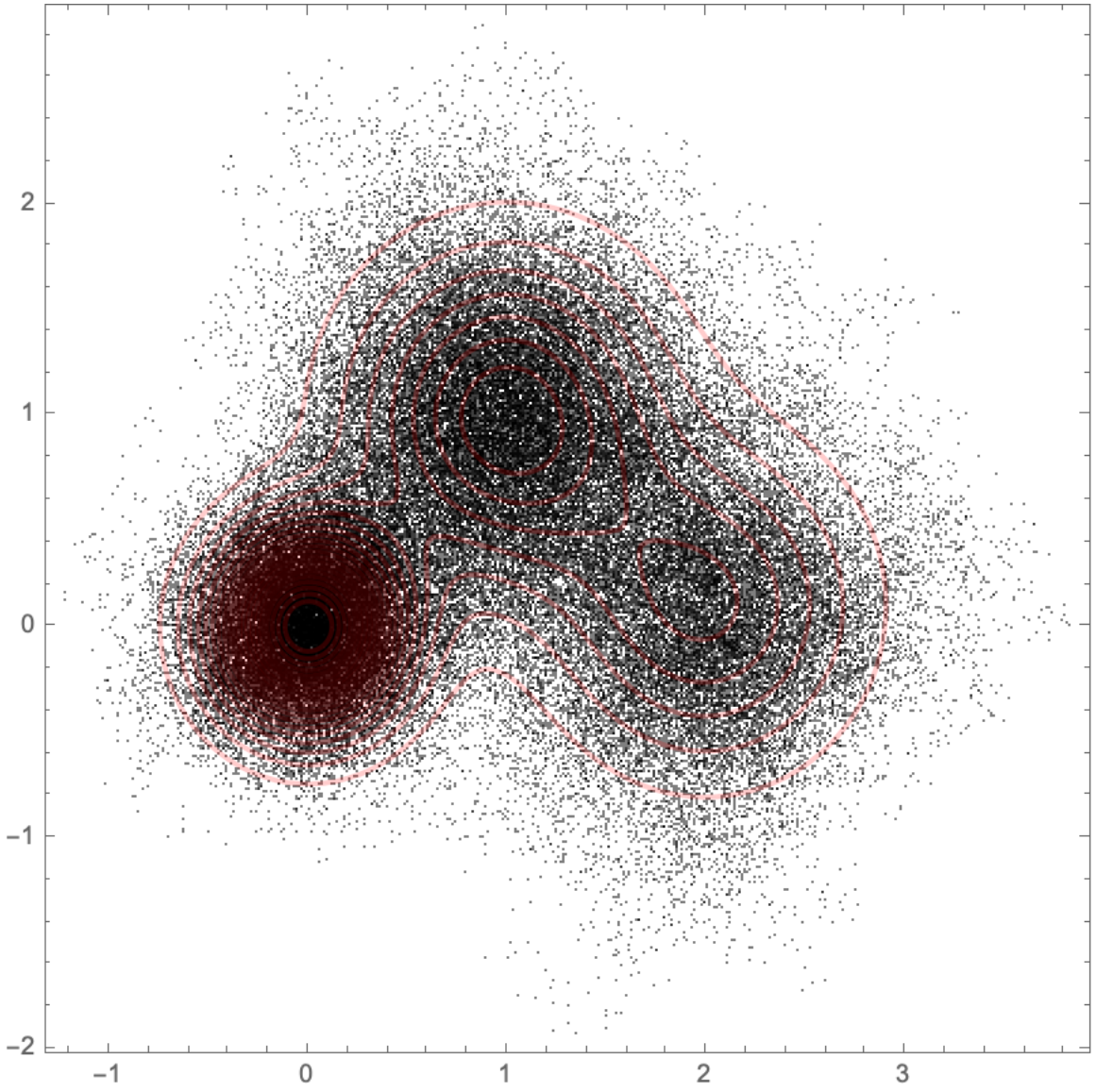


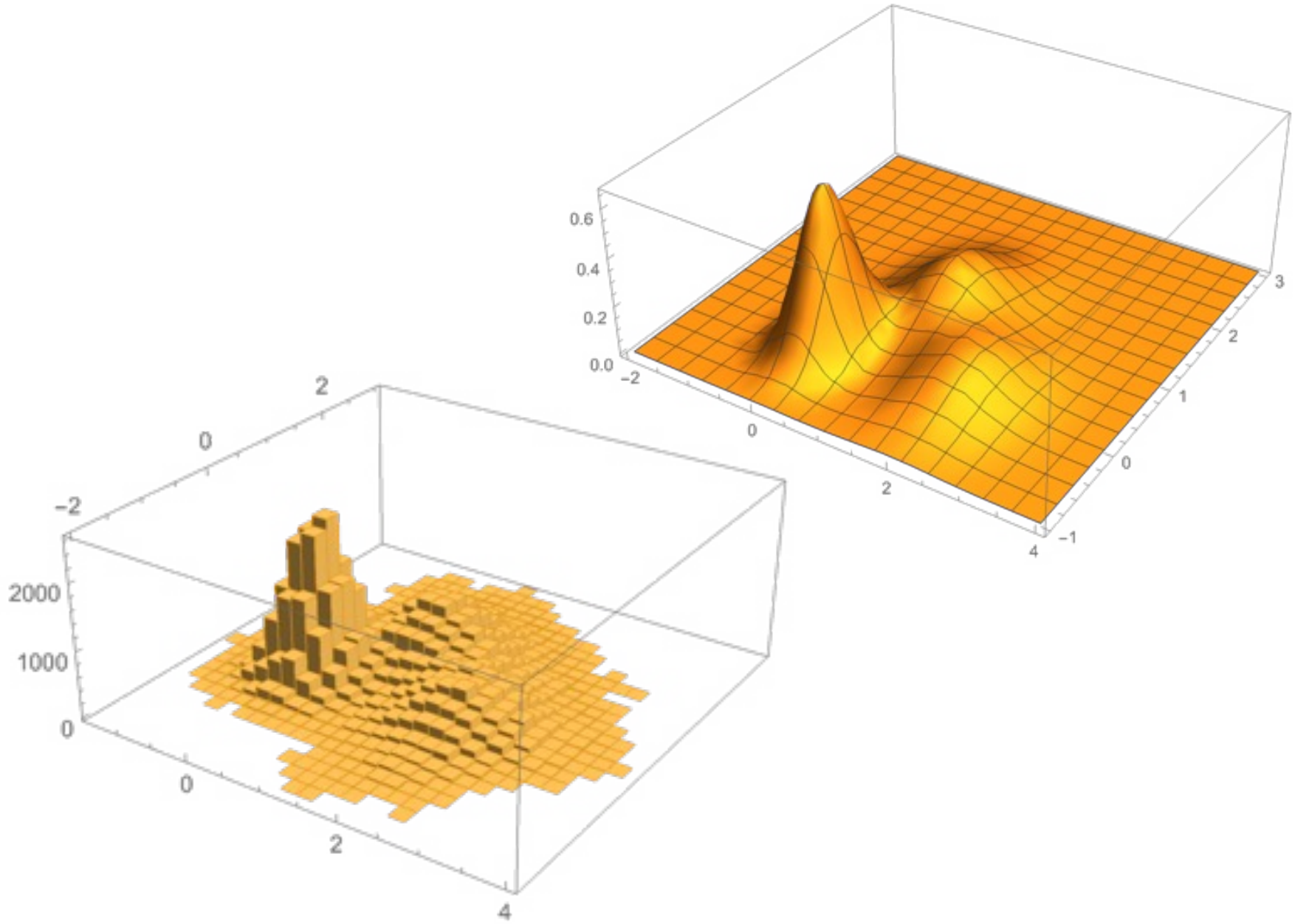


100000 steps

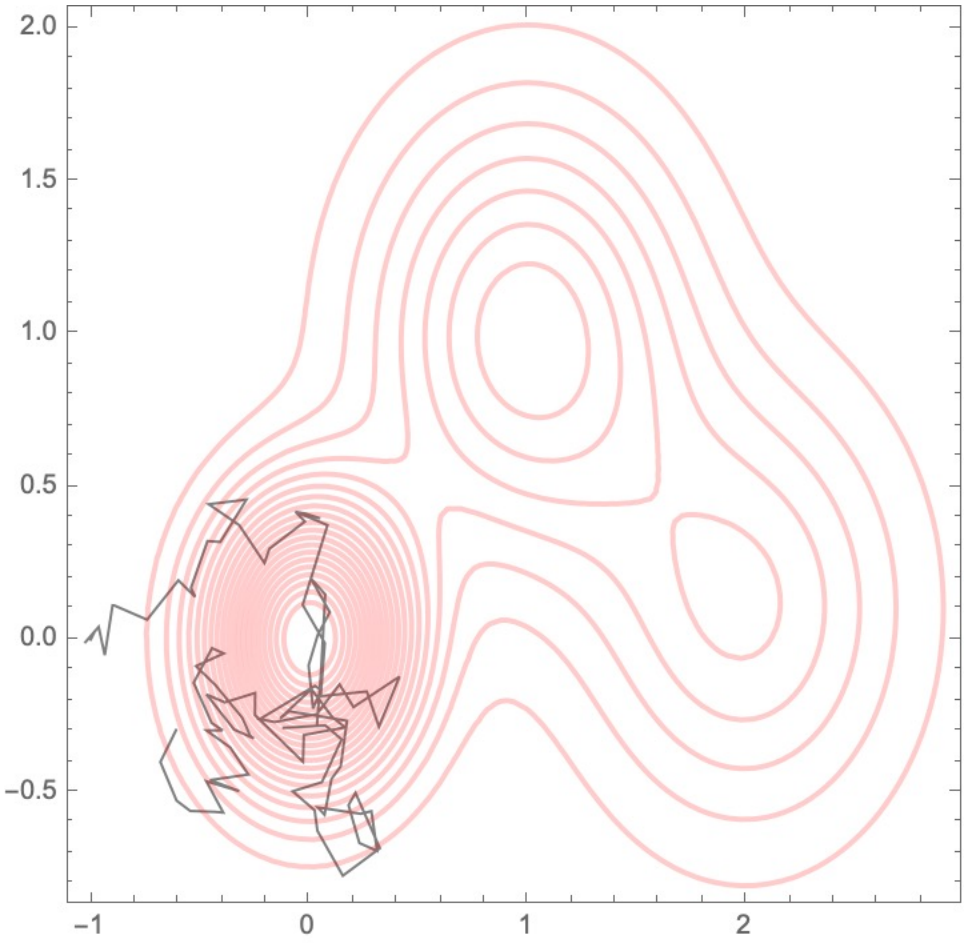


100000 steps

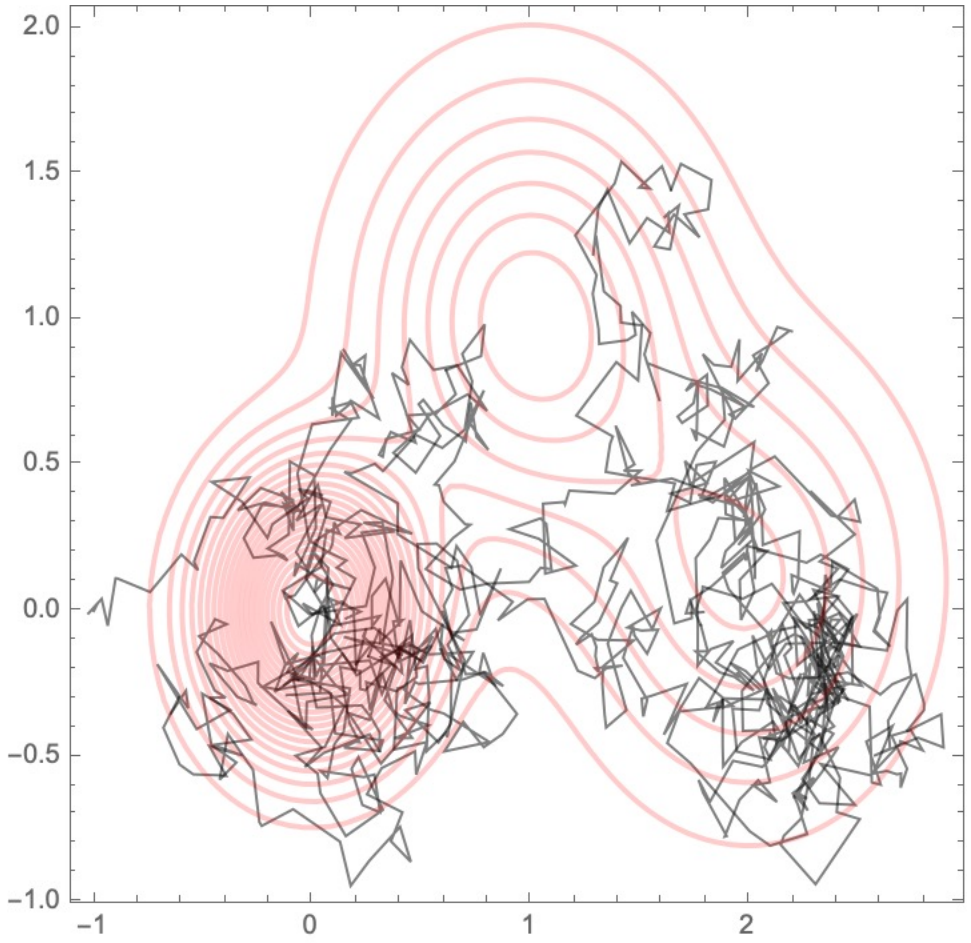



0.6
0.4
0.2
0.0
−2
0
2
3
2
1
0
−1
4
2
0
−2
2000
1000
0
0
2
4

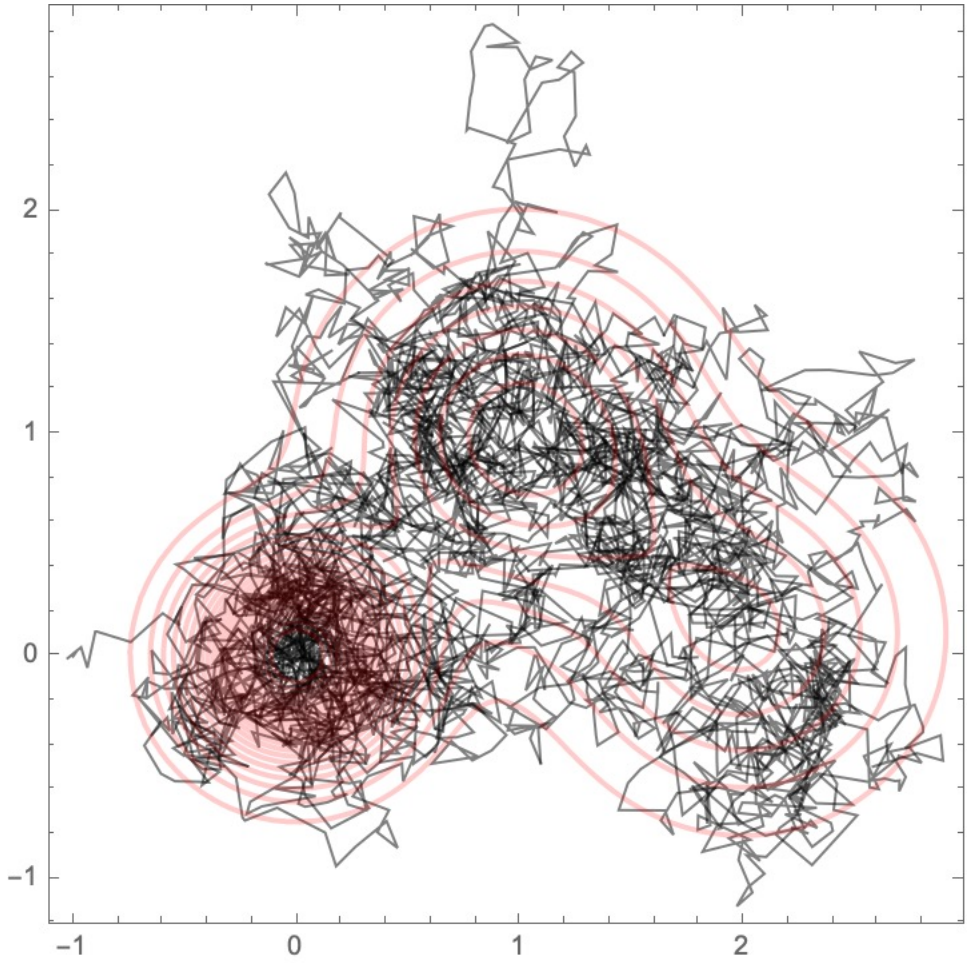
100 steps



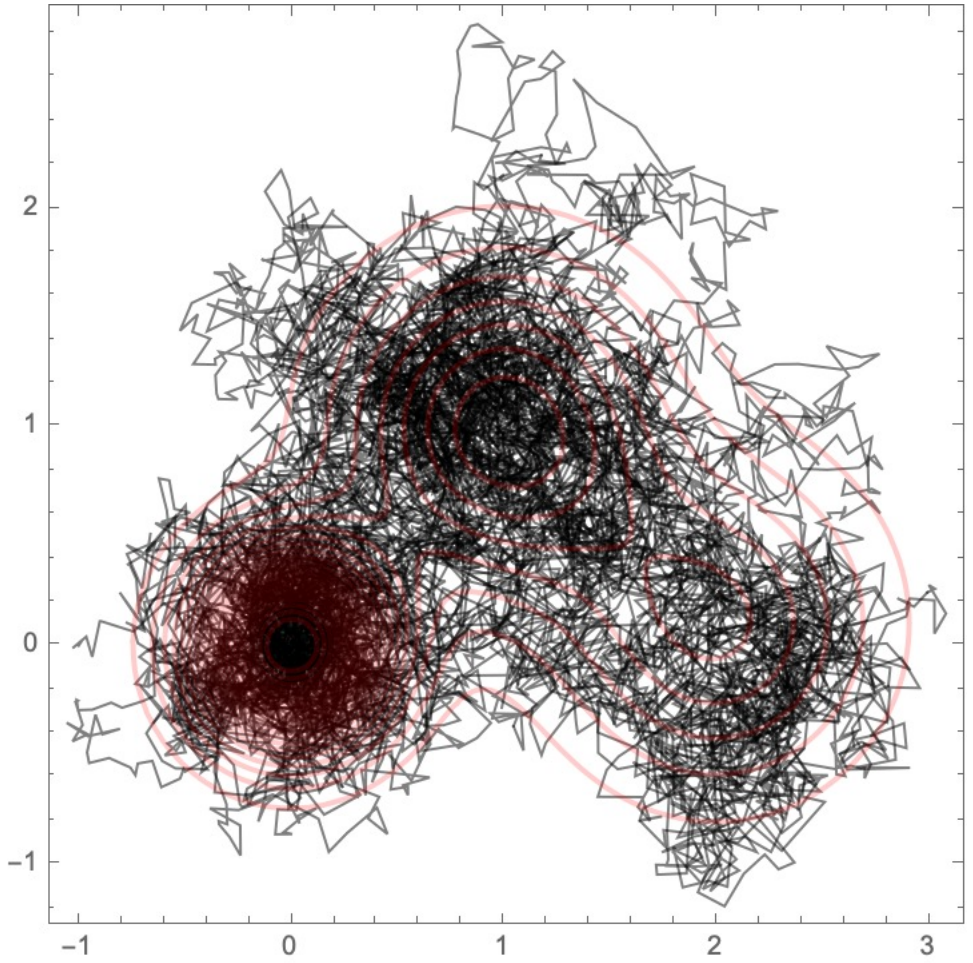
1000 steps



4000 steps



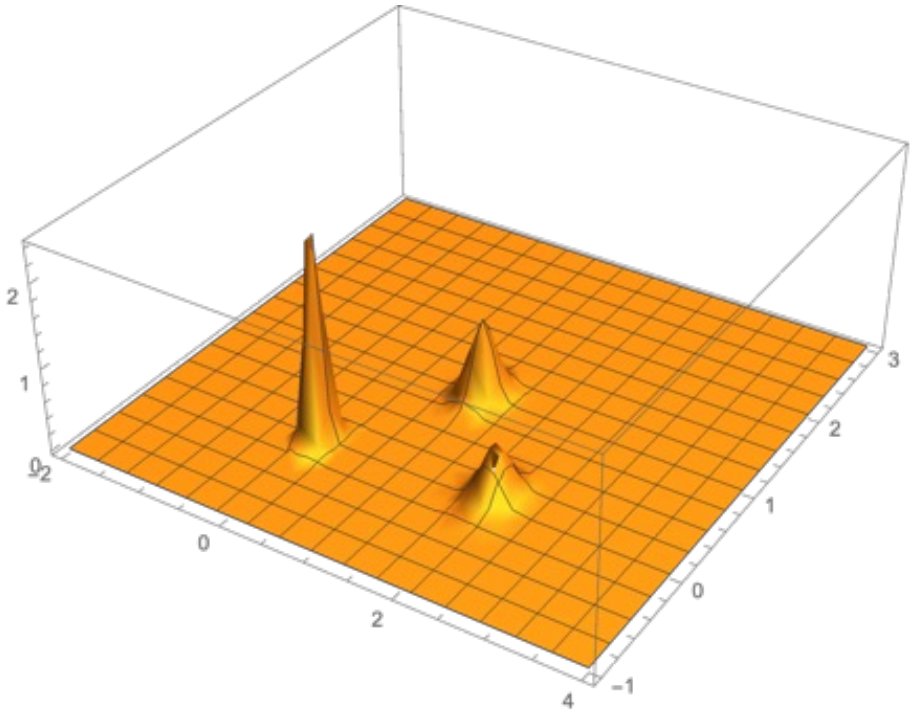
10000 steps



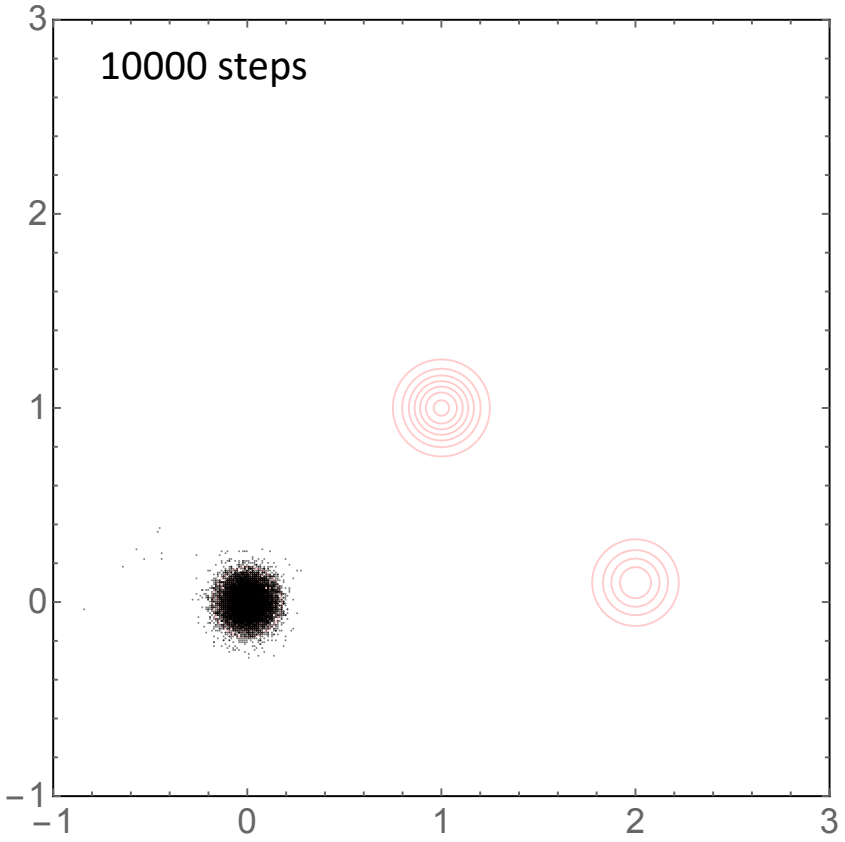
Notice that when the peaks are very narrow, the random walker may have problems visiting all of the peaks

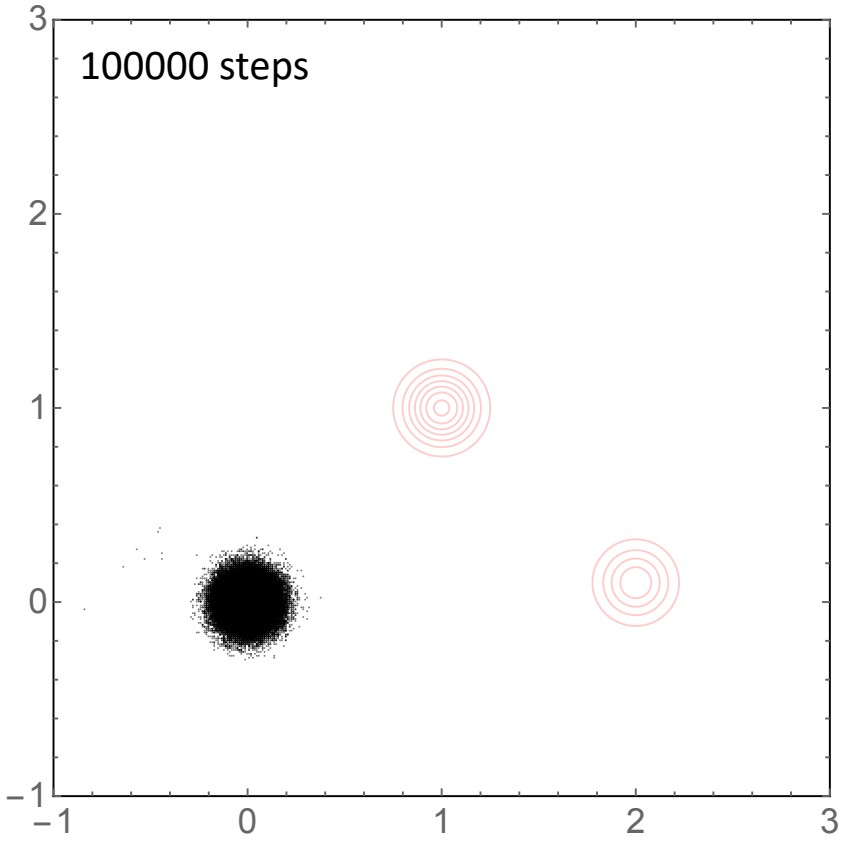
$$p(x,y) = \sum_{i=1}^{3} \frac{\alpha_i}{\sqrt{2\pi\sigma_i^2}} \exp\left[-\frac{(x-\mu_{x,i})^2 + (y-\mu_{y,i})^2}{2\sigma_i^2}\right]$$

$$\begin{array}{llll} \alpha_1 = 0.5; & \mu_{x,1} = 0; & \mu_{y,1} = 0; & \sigma_1 = 0.0725; \\ \alpha_2 = 0.3; & \mu_{x,2} = 1; & \mu_{y,2} = 1.; & \sigma_2 = 0.125; \\ \alpha_3 = 0.2; & \mu_{x,3} = 2; & \mu_{y,3} = 0.1; & \sigma_3 = 0.125; \end{array}$$

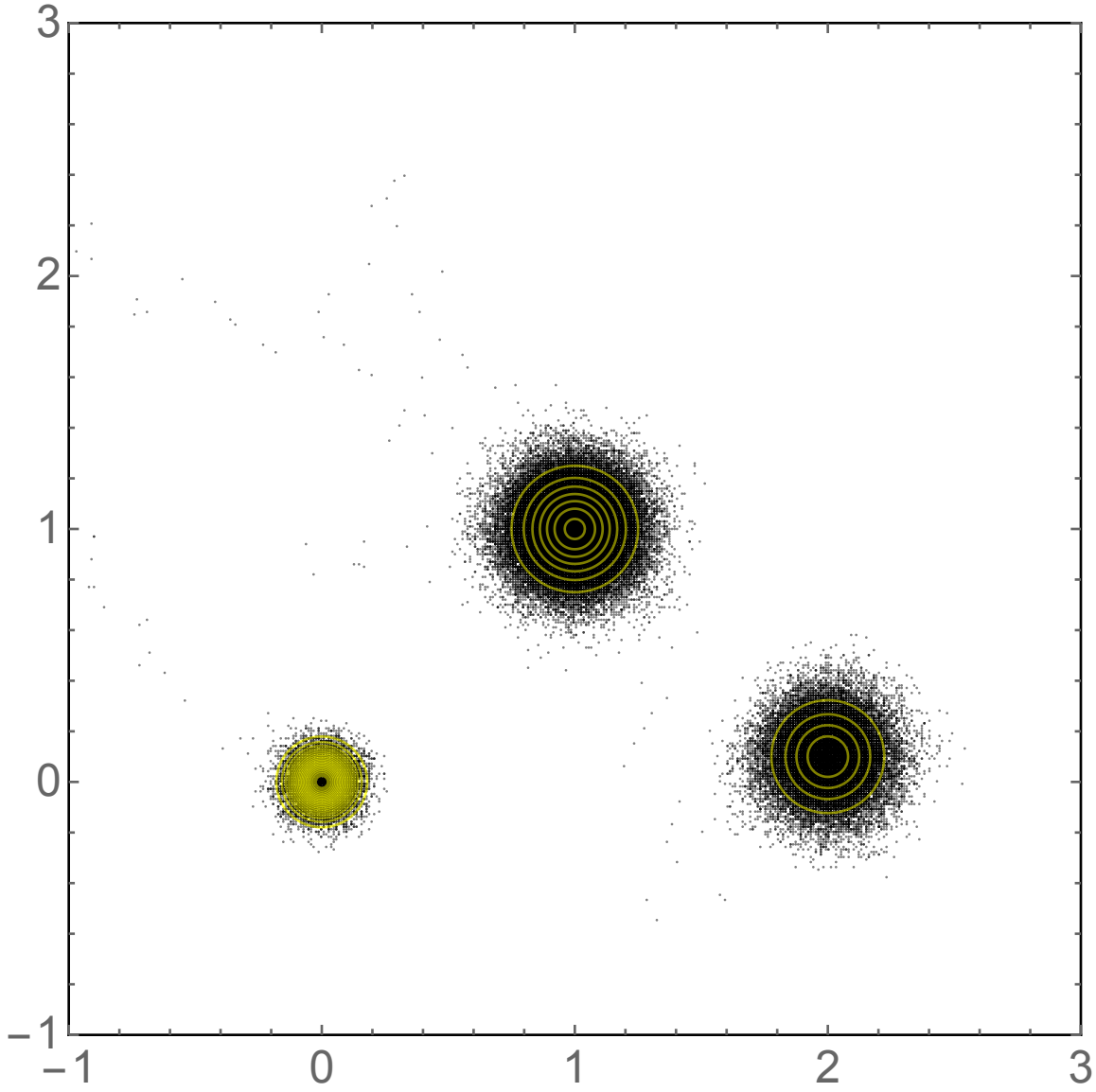


With isolated, narrow peaks, increasing the number of steps may not suffice



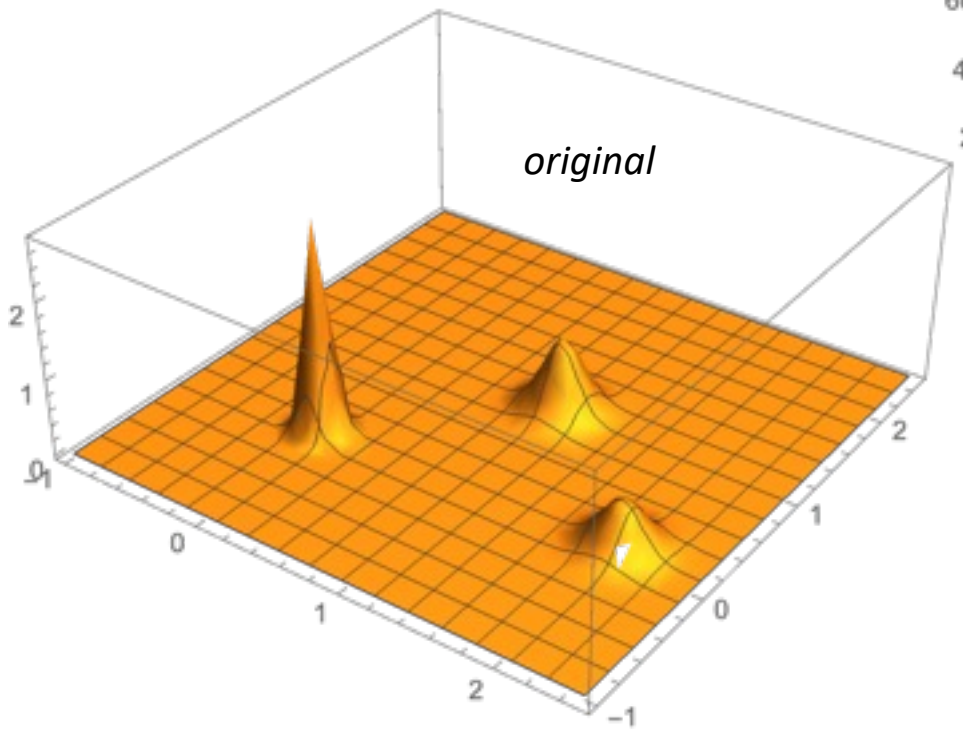


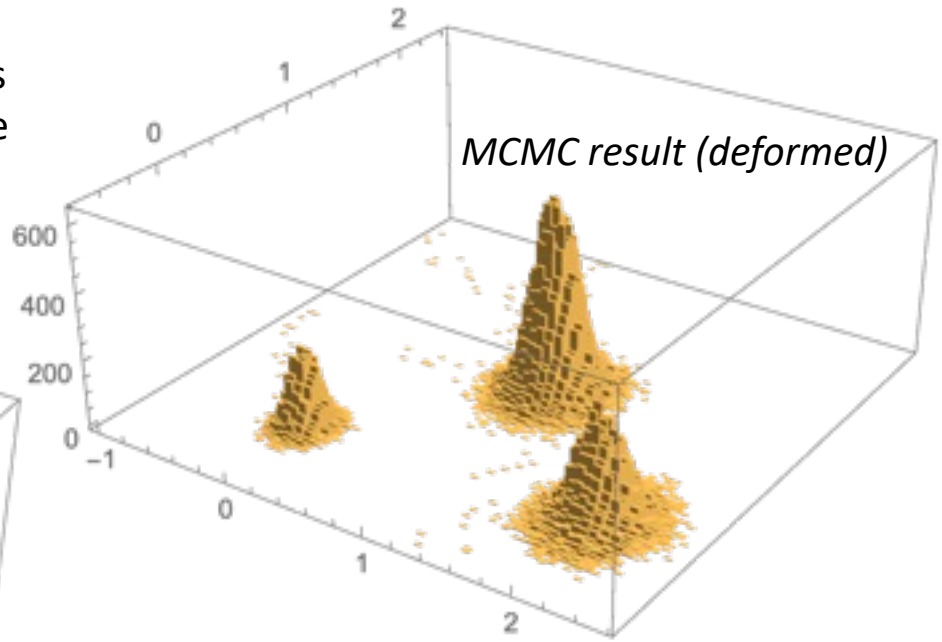
100000 steps, subdivided into 10 parallel chains with random starting points



The starting points of the chains are uniformly distributed in the plot region, however the "regions of influence" of each peak vary considerably.

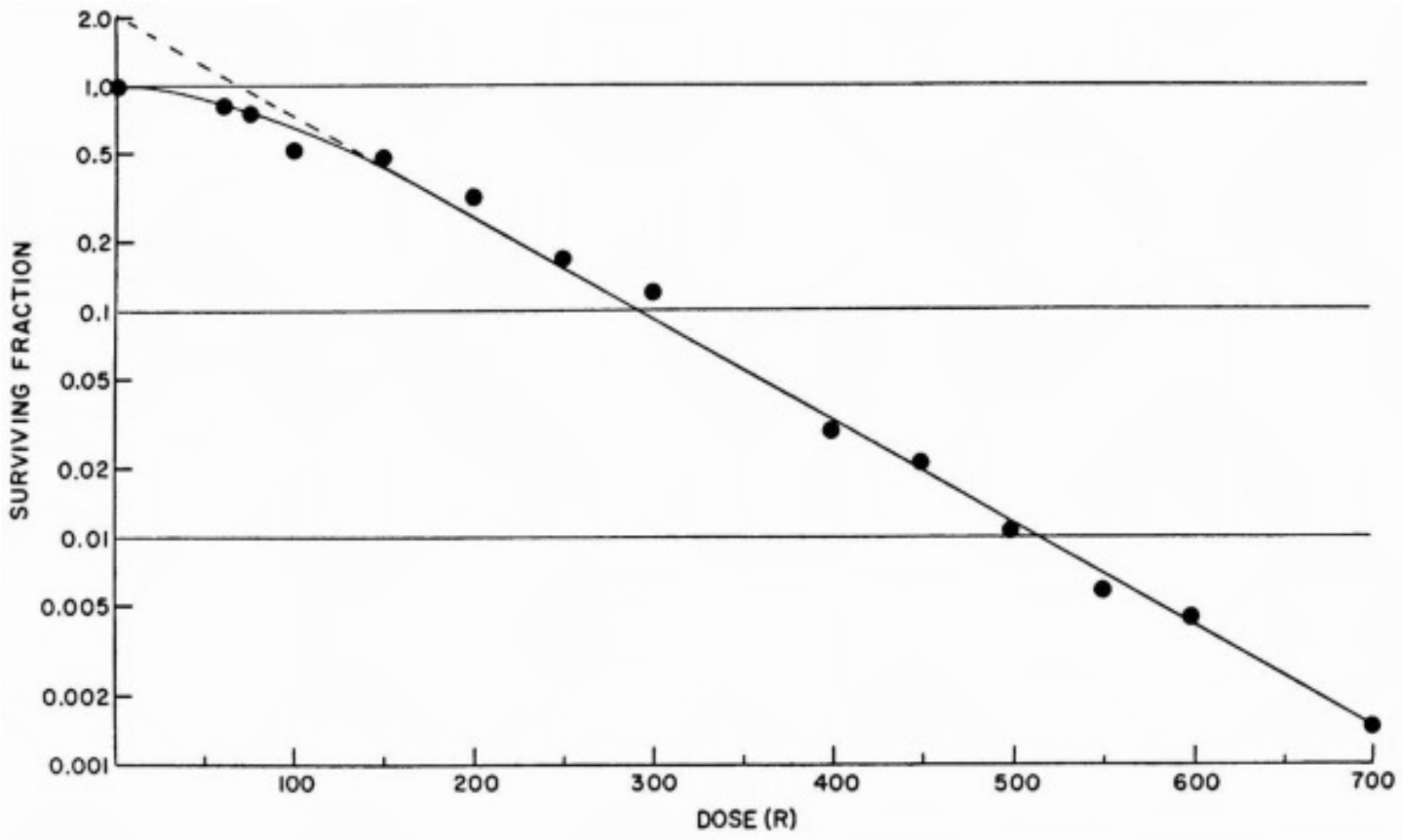
This leads to more chains being attracted into the lower peaks, with the result that the distribution is somewhat deformed (wrong alpha's in the mixture model)





Many techniques have been developed to avoid these pitfalls

Example of application of the MCMC technique in radiobiology



Survival curve for HeLa cells in culture exposed to x-rays. (From Puck TT, Markus Pl: Action of x-rays on mammalian cells. *J Exp Med* 103:653-666, 1956)

Phenomenology: the linear-quadratic law

$$S(D) \approx e^{-\alpha D - \beta D^2}$$

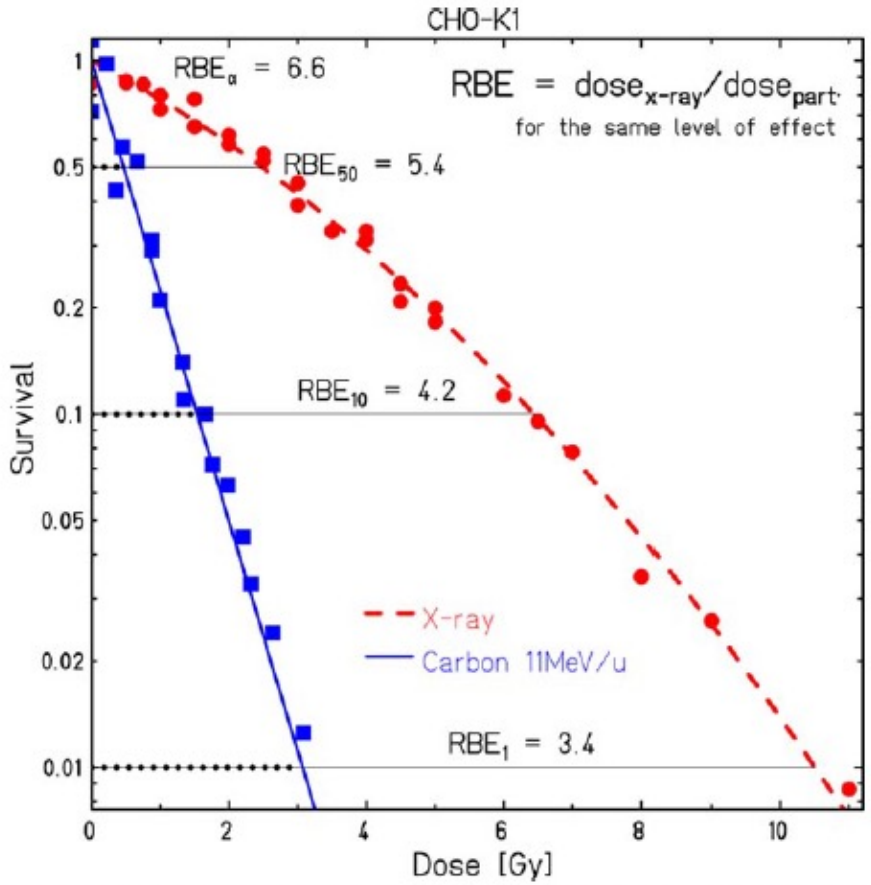


Fig. 1. Clonogenic survival curves illustrating the higher efficiency of the carbon ions compared with X-rays [10] (courtesy of the author, dr. Wilma K. Weyrather).

Example: Target theory

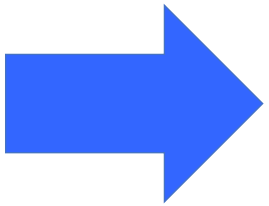
Simple Poisson model:

Probability of hitting *n* times a given target, when the average number of good hits is *a*:

$$P(n) = \frac{a^n}{n!} e^{-a}$$

Probability missing the target: $P(0) = e^{-a}$

Average number of hits: $a = D/D_0$



$$S(D) = P(0, D) = e^{-D/D_0}$$

Multitarget model, asymptotic behavior and threshold effect.

If there are multiple targets, say *n* targets, all of which must be hit to kill a cell, then the probability of missing at least one of them – i.e., the survival probability – is

$$S(D) = 1 - (1 - e^{-D/D_0})^n$$

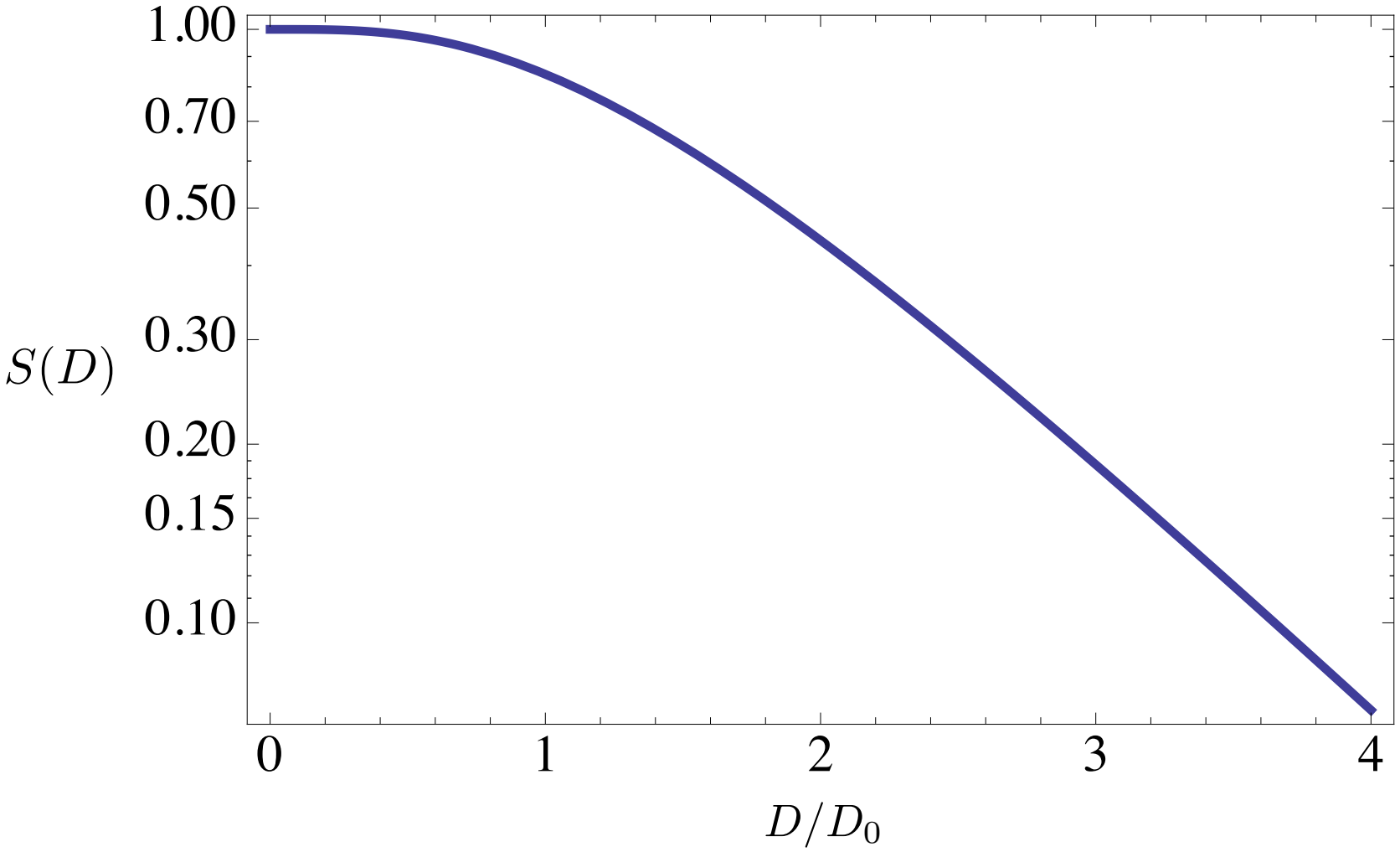
then, for large dose

$$S(D) \approx n e^{-D/D_0}$$

i.e.,

$$\ln S(D) \approx \ln n - D/D_0$$

which is a linear relation with intercept ln *n*, and slope -1/D_0.


1.00
0.70
0.50
0.30
0.20
0.15
0.10
$S(D)$
0
1
2
3
4
D/D_0

Notice that

$$\left[\frac{d}{dD}e^{-\alpha D-\beta D^2}\right]_{D=0} = (-\alpha - 2\beta D)e^{-\alpha D-\beta D^2}\Big|_{D=0} = -\alpha$$

and that

$$\frac{d}{dD}\left[1-(1-e^{-D/D_0})^n\right]_{D=0} = -n\,\frac{e^{-D/D_0}}{D_0}(1-e^{-D/D_0})^{n-1}\Big|_{D=0} = 0$$

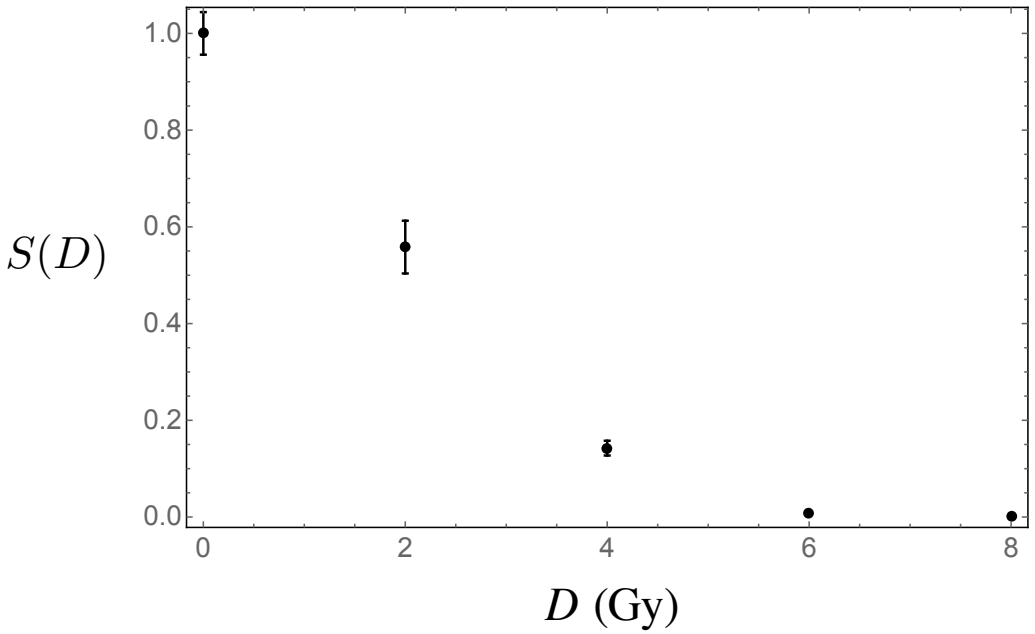
The derivatives differ in the origin, and the multitarget model fails to reproduce the observed linear-quadratic law.

The RCR (Repairable-Conditionally Repairable Damage) model

In this case the surviving fraction is

$$S = \exp(-aD) + bD\exp(-cD)$$

This is a 3-parameter expression, which is not easy to fit to data when the data set is small.



1a. Simple Gaussian likelihood for the LQ model

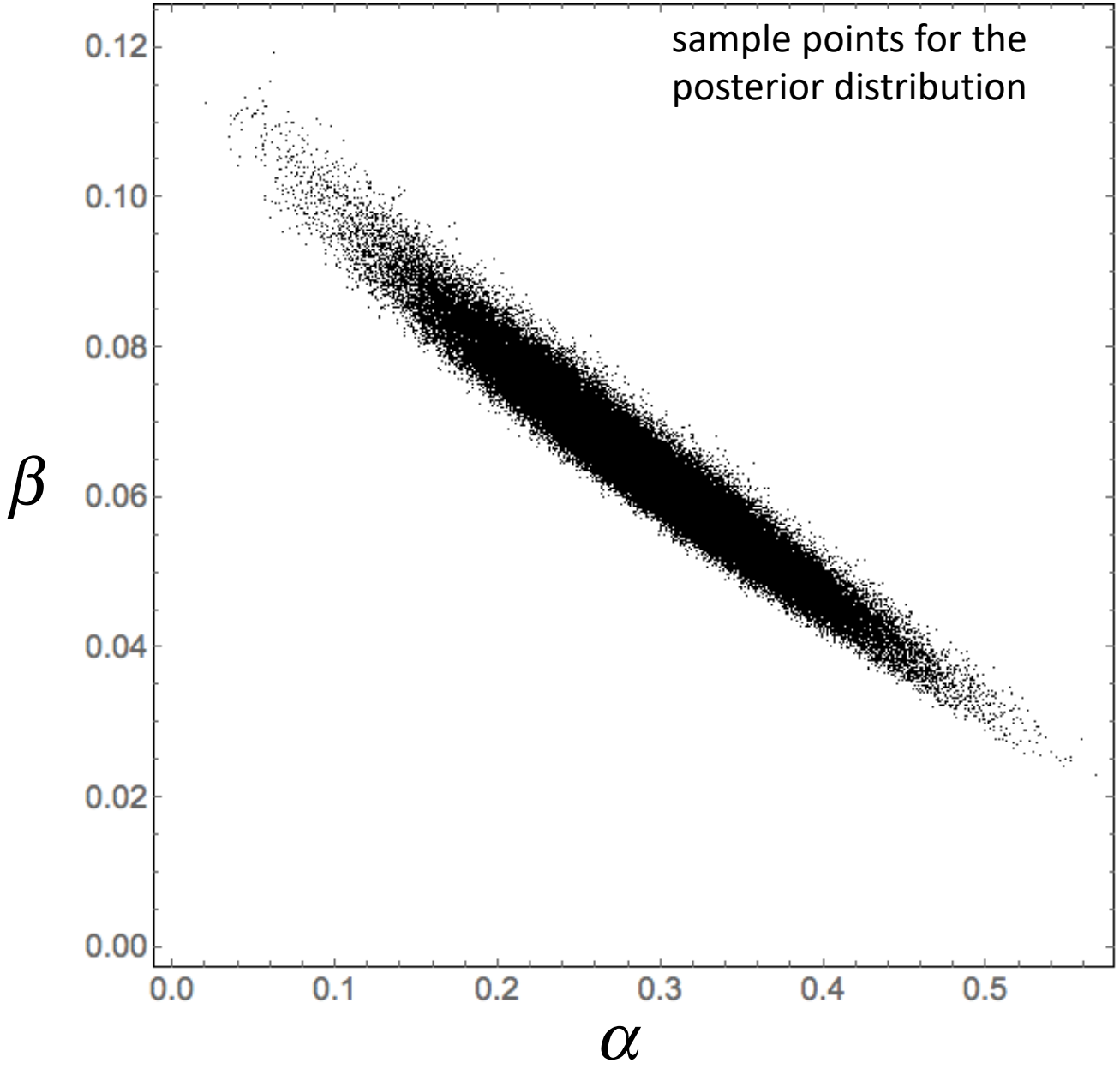
$$L(\alpha,\beta)=\prod_k \exp\left(-\frac{\left(S_k-S(\alpha,\beta)\right)^2}{2\sigma_k^2}\right)$$

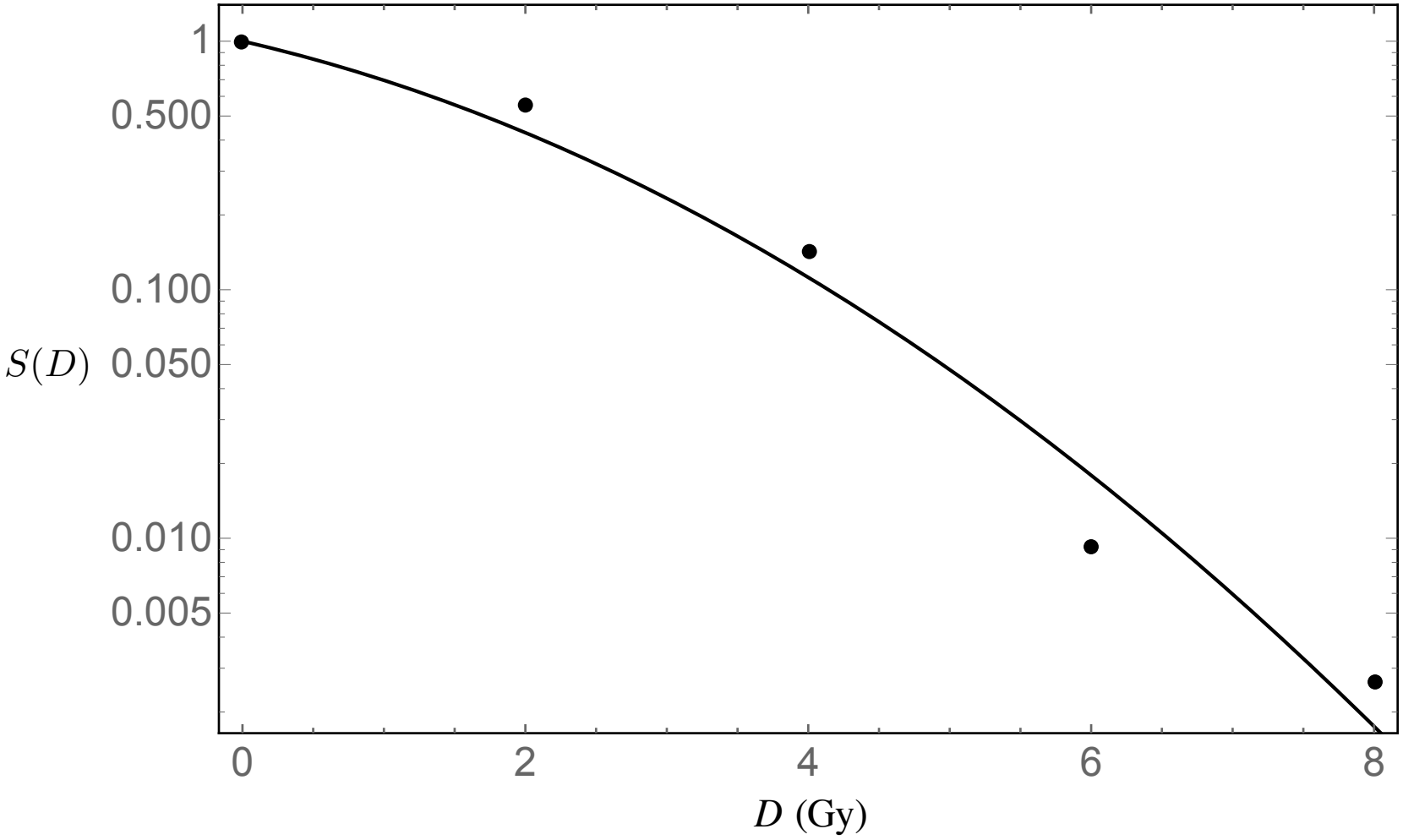
1b. Chose exponential priors for the parameters

1c. Complete posterior pdf

$$p\left(\alpha,\beta|\{S_k\},I\right)=\left[\prod_k \exp\left(-\frac{\left(S_k-S(\alpha,\beta)\right)^2}{2\sigma_k^2}\right)\right]\exp(-0.1\alpha)\exp(-0.1\beta)$$

1d. Use MCMC to find the MAP estimate (and any moment of the pdf)


sample points for the
posterior distribution
0.12
0.10
0.08
0.06
0.04
0.02
0.00
β
0.0
0.1
0.2
0.3
0.4
0.5
α


1
0.500
0.100
0.050
0.010
0.005
$S(D)$
0
2
4
6
8
D (Gy)

2a. Simple Gaussian likelihood for the RCR model

$$L(a,b,c) = \prod_k \exp\left(-\frac{\left(S_k - S(a,b,c)\right)^2}{2\sigma_k^2} \right)$$

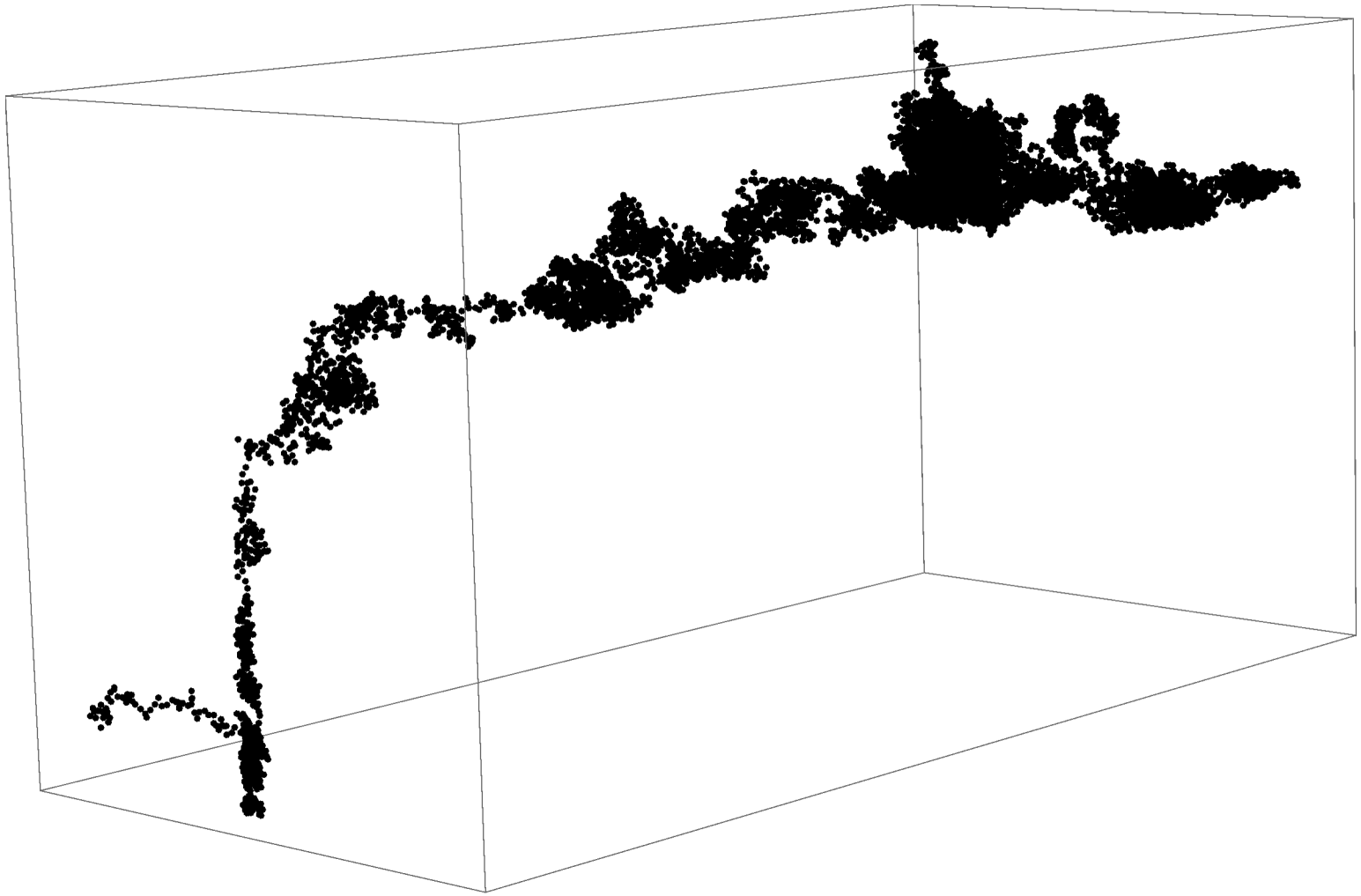
2b. Chose exponential priors for the parameters

2c. Complete posterior pdf

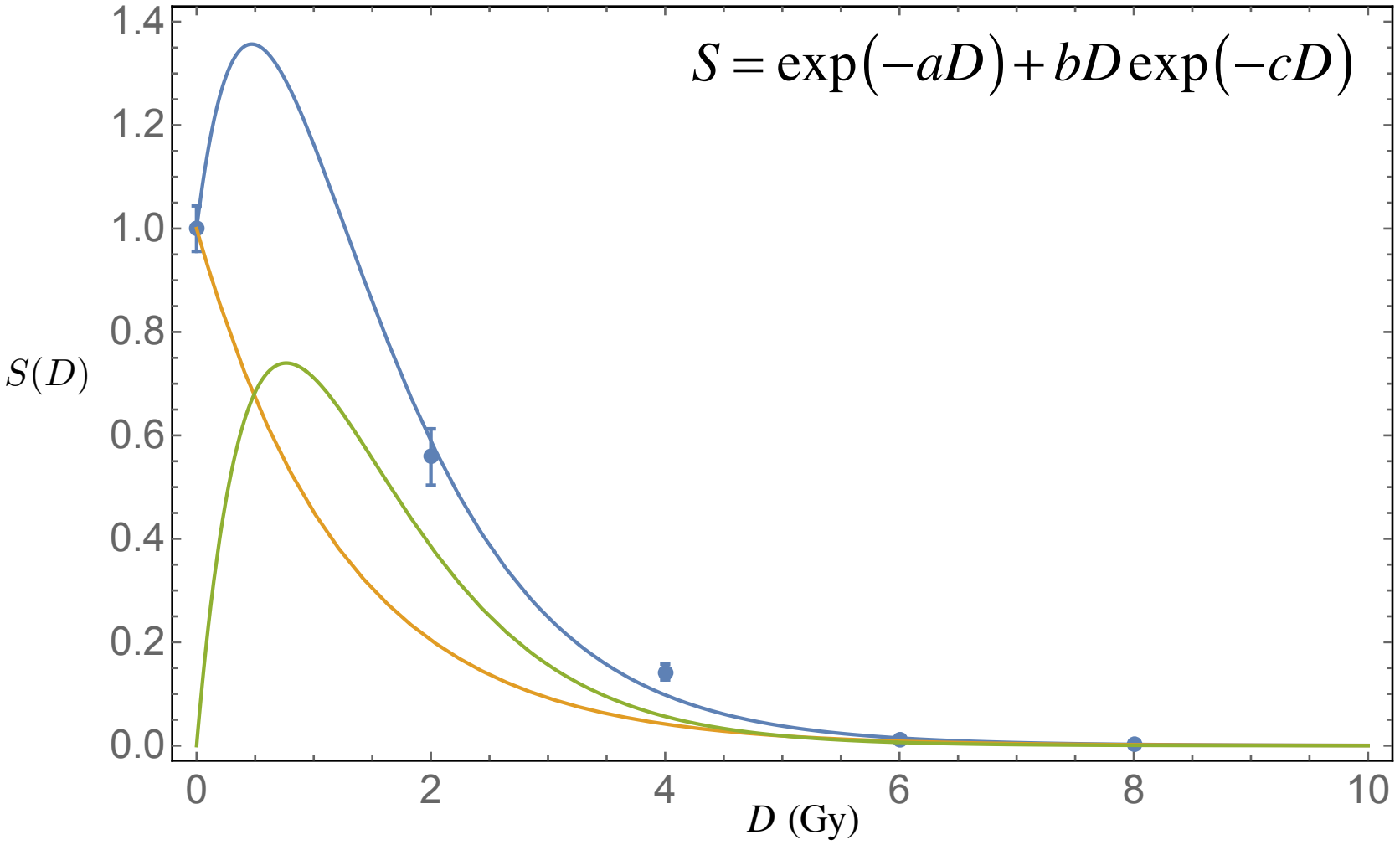
$$p\left(a,b,c \middle| \{S_k\}, I\right) = \left[\prod_k \exp\left(-\frac{\left(S_k - S(a,b,c)\right)^2}{2\sigma_k^2} \right) \right] e^{-0.2a} e^{-0.2b} e^{-0.2c}$$

2d. Use MCMC to find the MAP estimate (and any moment of the pdf)

Path in (a,b,c) space

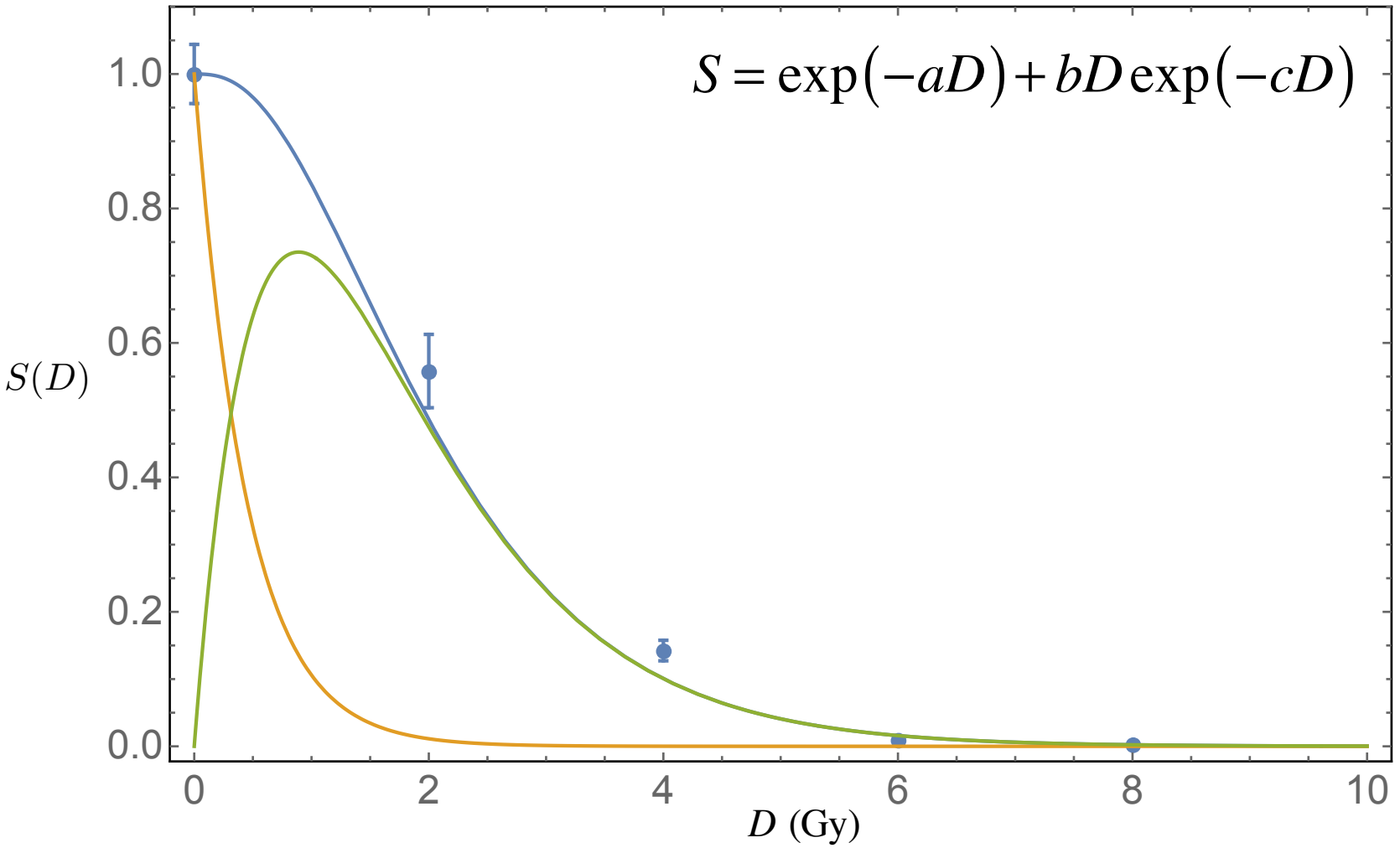


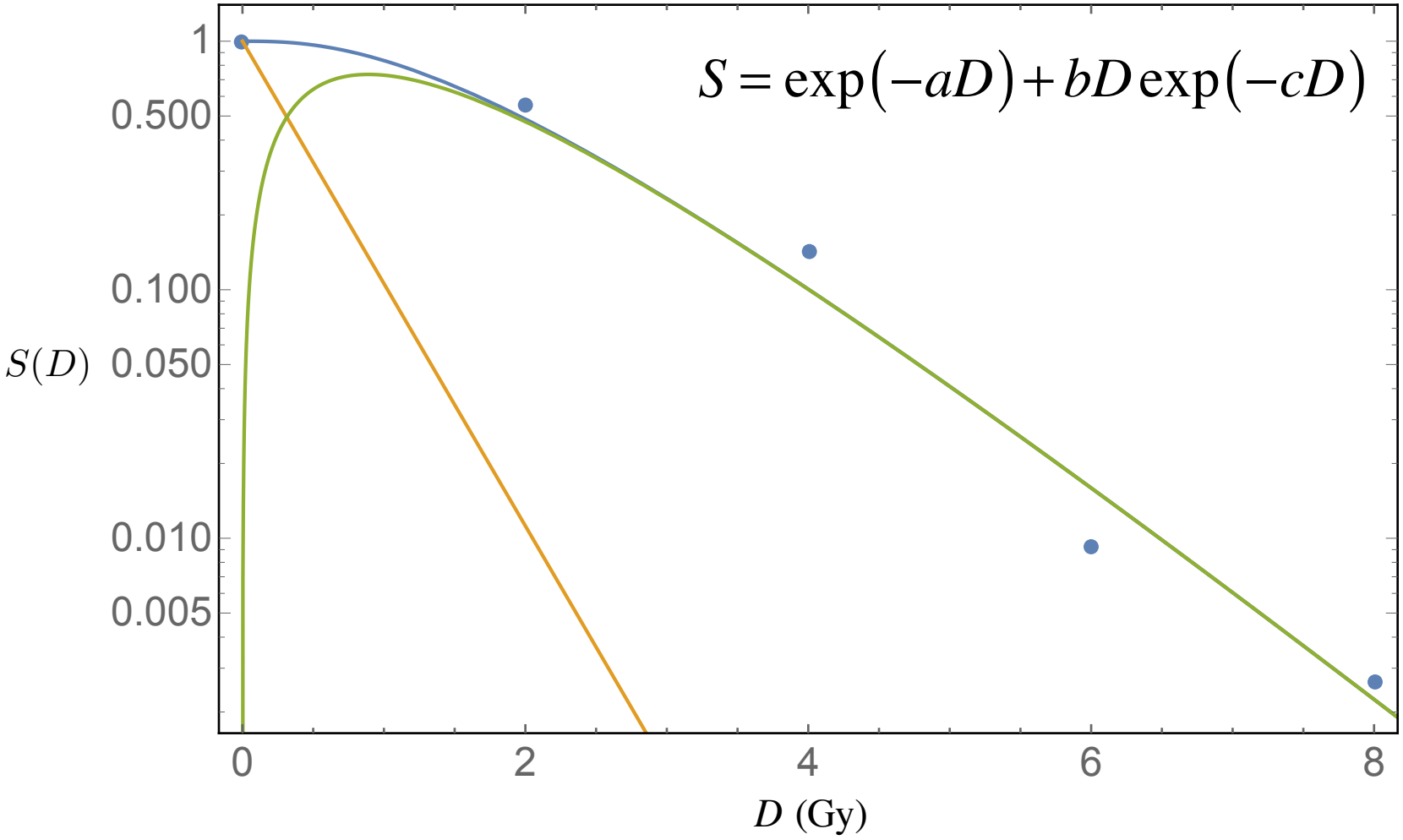
Fit showing individual components: unsatisfactory result



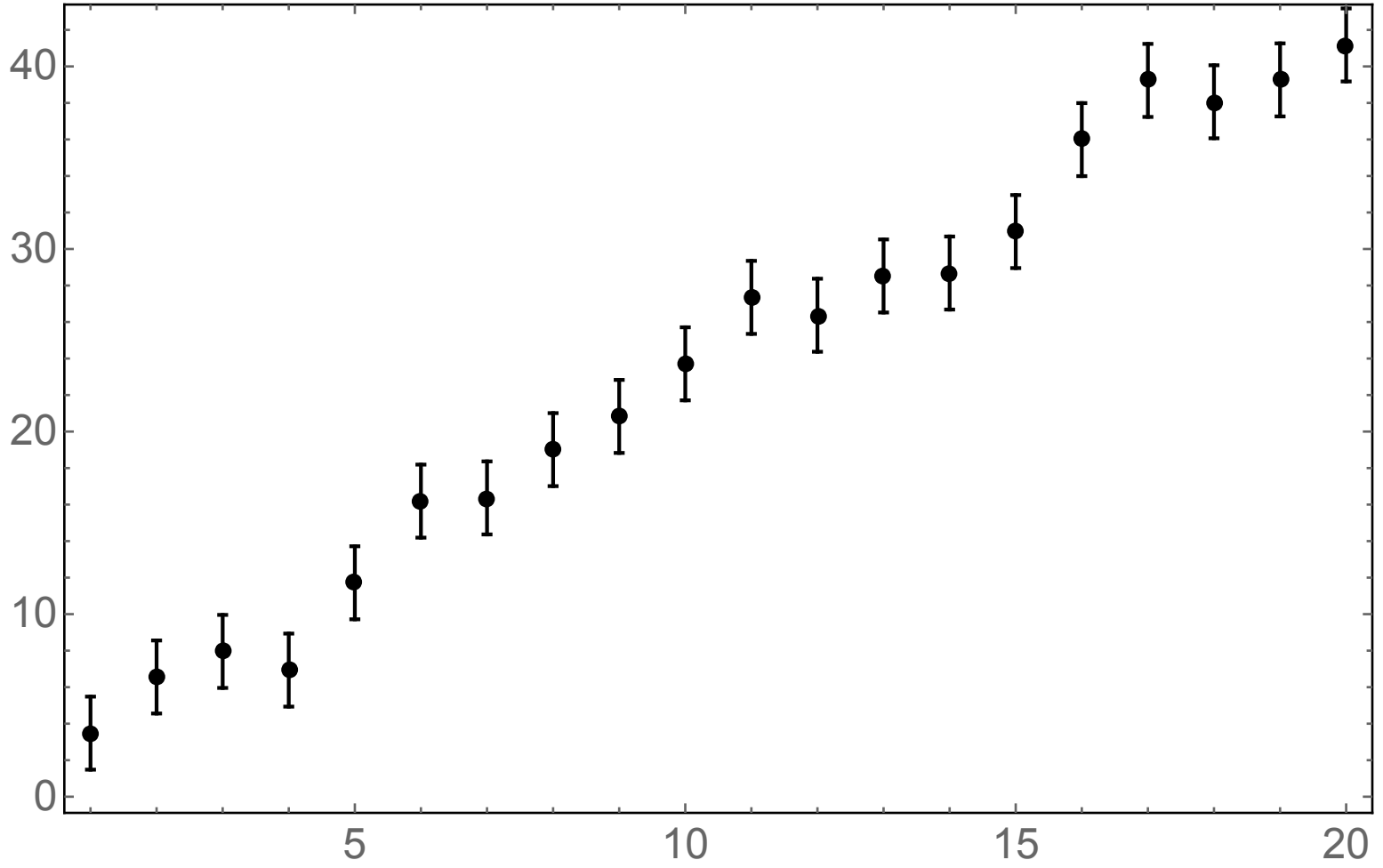
Revise priors to include constraint on derivative

(priors vanish where derivative in the origin is positive)

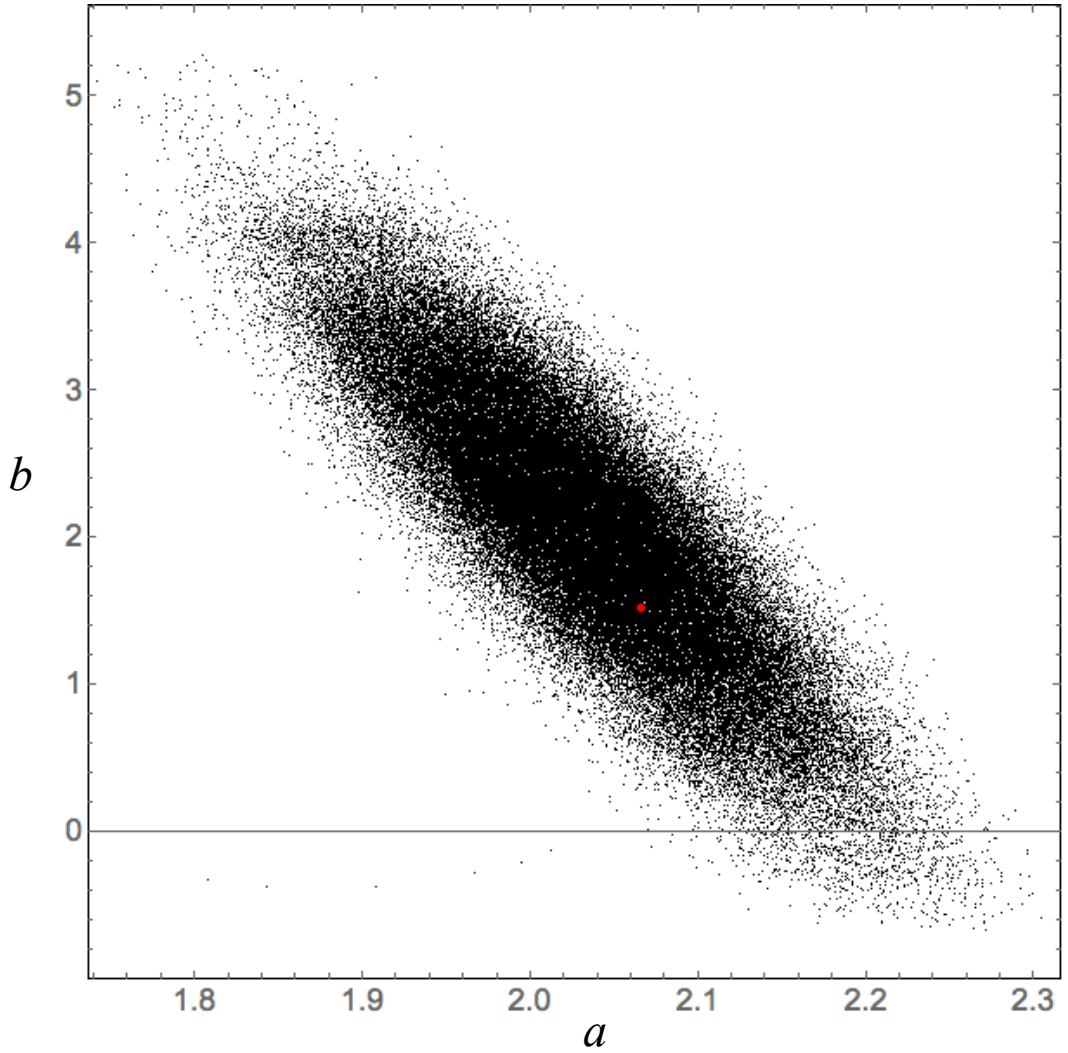


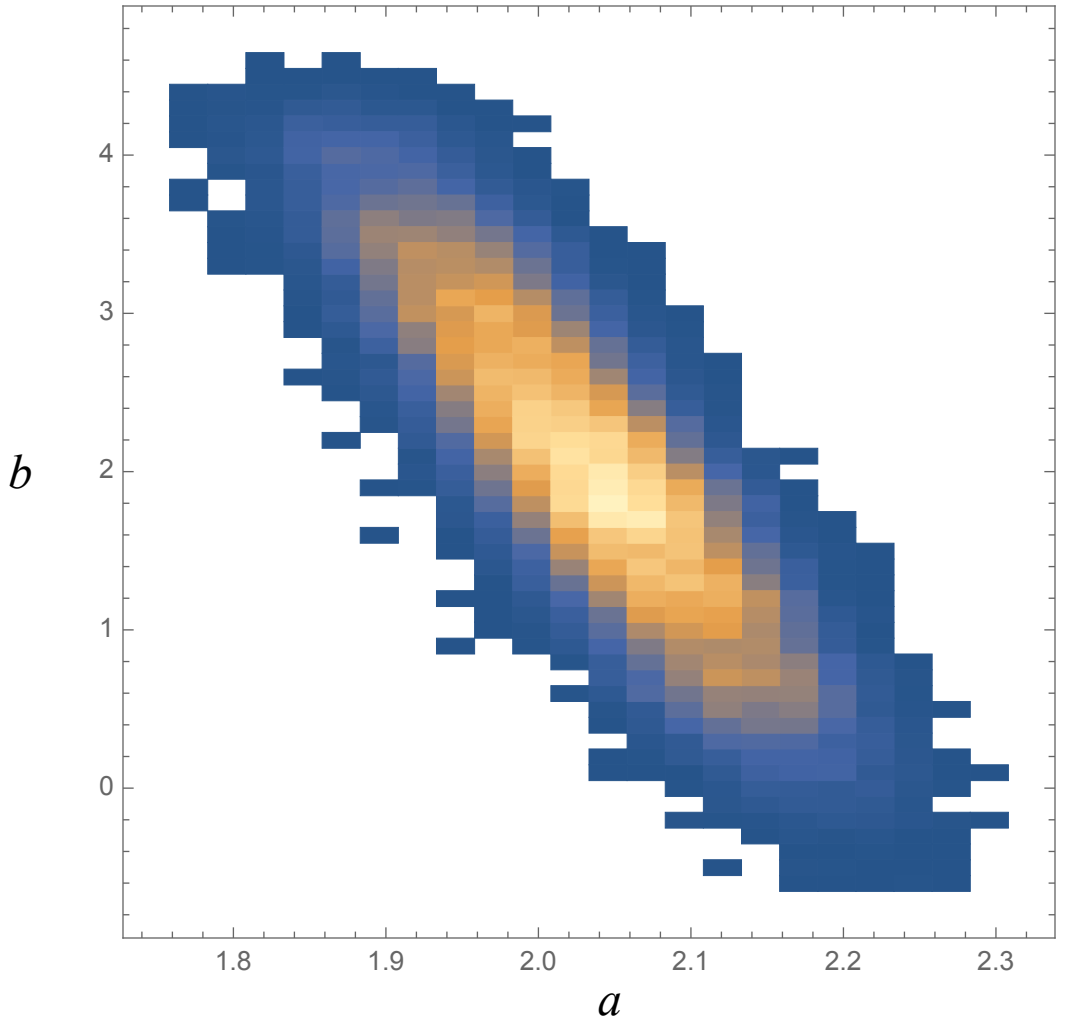

$S = \exp(-aD) + bD\exp(-cD)$
1
0.500
0.100
0.050
0.010
0.005
$S(D)$
0
2
4
6
8
D (Gy)

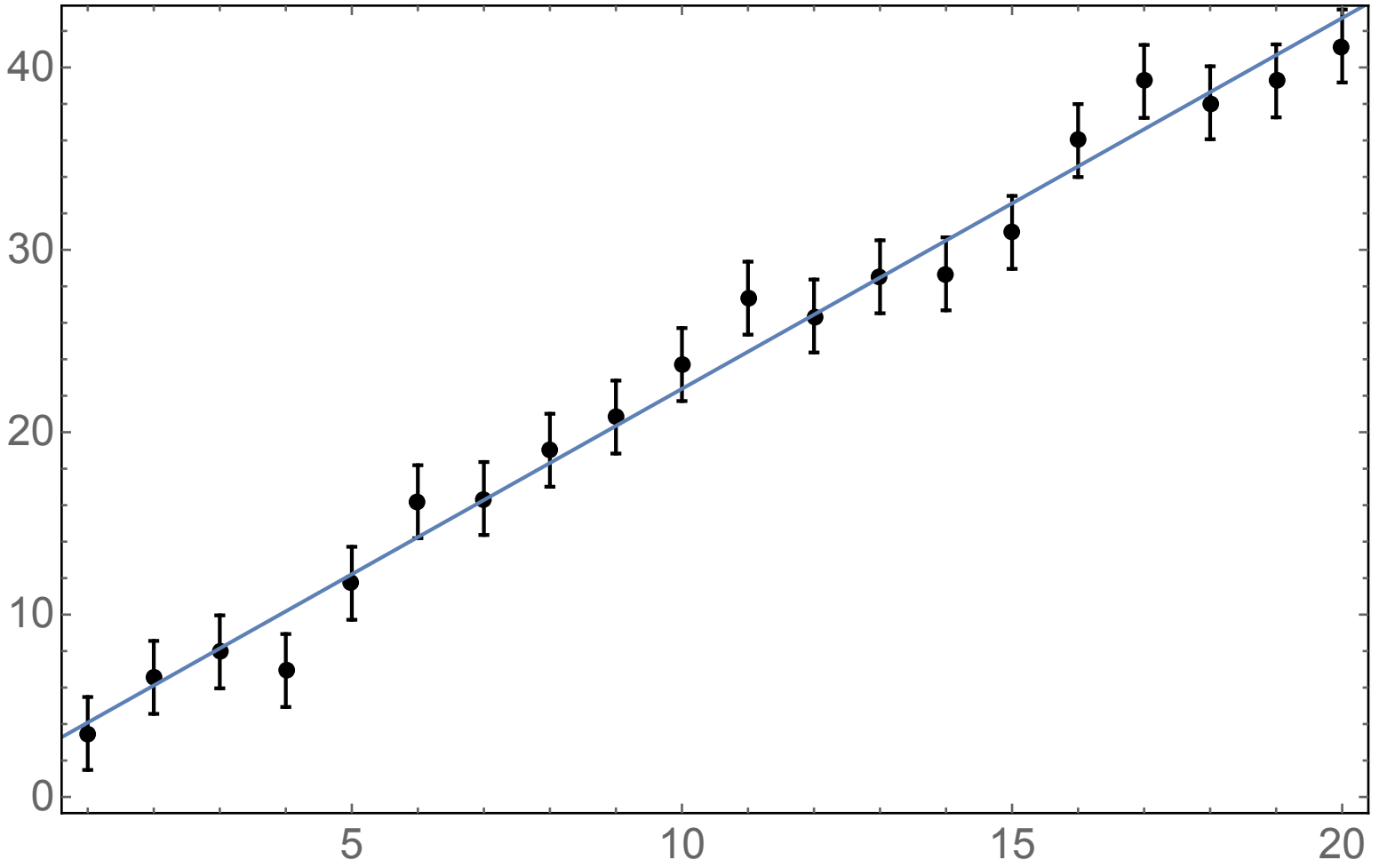
“Straight line fit” with the MCMC
An example with Gaussian errors and exponential priors.



model $y = ax + b$

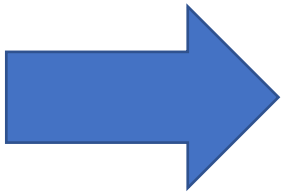



b
4
3
2
1
0
1.8
1.9
2.0
2.1
2.2
2.3
a


40
30
20
10
0
5
10
15
20

MCMC software

Course Webpage



Useful links

- BLIP (Bayesians Laboring In Physics)
- BUGS Project (Bayesian Inference Using Gibbs Sampling) + WinBUGS + OpenBUGS
- EMCEE affine invariant MCMC documentation
- International Society for Bayesian Analysis
- JAGS (Just Another Gibbs Sampler)
- MacMCMC
- Webpage of Larry Bretthorst
- Webpage of Tom Loredo
- Stan
- Statistics bibliography at SLAC
- Valencia meetings
- Wikipedia (article on Rev. T. Bayes)

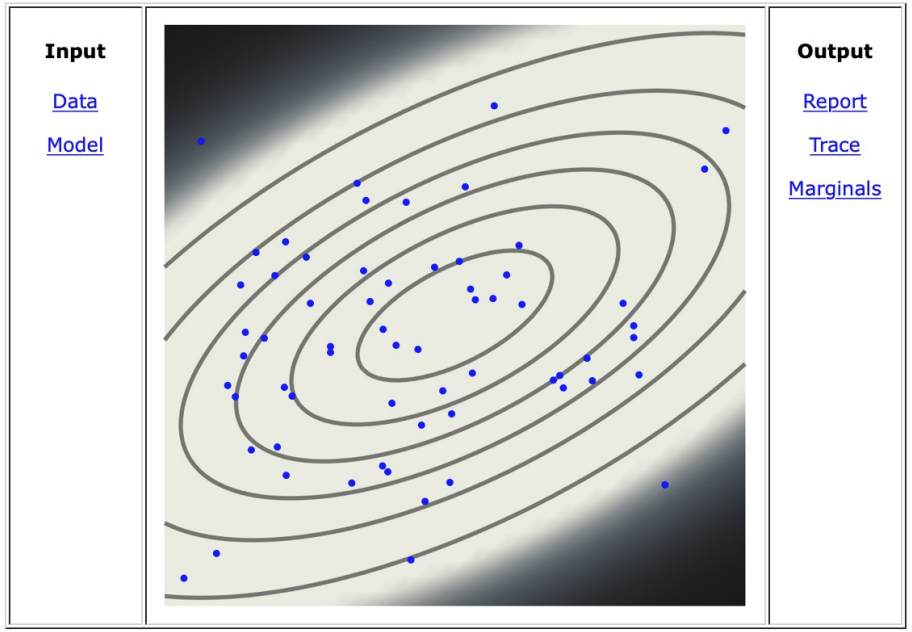
MacMCMC (v1.5)

State-of-the-art Data Analysis for Mac OS X™

MacMCMC is a free and extremely powerful application for the analysis of data of any kind. It is one half of a two-part project. The other half is a free ebook—a strongly recommended preliminary—available here.

To see *MacMCMC* in action, consider this famous example from the literature (Arnold and Libby, 1949):

Carbon-14 Dating



Given the *MacMCMC* report, any graphing software may be used to prepare a plot showing model versus data.
Note: The blue line in this plot uses mean estimates; the red line shows the prior uncertainty for parameter A.

Principal Features

General

- Complete, standalone Mac application
- 100% Bayesian inference
- 100% ensemble MCMC
- Access to low-level options
- Parallelized for maximum speed

https://causascientia.org/software/MacMCMC/MacMCMC.html

https://docs.pymc.io

Friendly modelling API

PyMC3 allows you to write down models using an intuitive syntax to describe a data generating process.

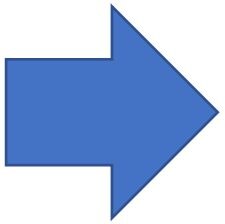
Cutting edge algorithms and model building blocks

Fit your model using gradient-based MCMC algorithms like NUTS, using ADVI for fast approximate inference — including minibatch-ADVI for scaling to large datasets — or using Gaussian processes to build Bayesian nonparametric models.

```
import pymc3 as pm

X, y = linear_training_data()
with pm.Model() as linear_model:
    weights = pm.Normal("weights", mu=0, sigma=1)
    noise = pm.Gamma("noise", alpha=2, beta=1)
    y_observed = pm.Normal(
        "y_observed",
        mu=X @ weights,
        sigma=noise,
        observed=y,
    )

    prior = pm.sample_prior_predictive()
    posterior = pm.sample()
    posterior_pred = pm.sample_posterior_predictive(posterior)
```



practical demo with an MCMC program (MacMCMC)